Physics-Informed Neural Network-Based Parametric Deep Material Network for Multiphysics Behavior Prediction of Heterogeneous Materials with a Varying Morphology

Tianyi Li

Physics-Informed Neural Network-Based Parametric Deep Material Network for Multiphysics Behavior Prediction of Heterogeneous Materials with a Varying Morphology

Tianyi Li[a]

[a]*Dassault Systèmes, 10 rue Marcel Dassault, 78140 Vélizy-Villacoublay, France*

Abstract

Deep Material Network (DMN) has recently emerged as a data-driven surrogate model for heterogeneous materials. Given a particular microstructural morphology, the effective linear and nonlinear behaviors can be successfully approximated by such physics-based neural-network like architecture. In this work, a novel parametric DMN architecture is proposed for multiscale materials with a varying microstructure characterized by several parameters. A Physics-Informed Neural Network (PINN) is used to account for the dependence of DMN fitting parameters on the microstructural ones. Micromechanical constraints are prescribed both on the network architecture and on the output of this PINN. The proposed PINN-DMN architecture is also recast in a multiphysics setting, where physical properties other than the mechanical ones can also be predicted. In the numerical simulations conducted on three parametric microstructures, PINN-DMN demonstrates satisfying interpolative and extrapolative generalization capabilities when morphology varies. The effective multiphysics behaviors of such parametric multiscale materials can thus be predicted and encoded by PINN-DMN with high accuracy and efficiency.

Keywords: Deep material network, Parametric microstructures, Neural network, Machine learning, Effective multiphysics properties, Inverse identification

1. Introduction

The effective macroscopic behaviors of heterogeneous materials can be predicted by the physical properties of the constituents and the underlying microstructures using a multiscale modeling approach [1, 2, 3]. Among such models, mean-field homogenization schemes rely on the micromechanical assumptions (inclusion shapes, interaction between the constituents) of the microstructure being considered and are frequently used to predict linear and nonlinear behaviors of specific composites (such as the fiber-reinforced plastics) [4, 5]. Thanks to their analytical nature, they are computationally efficient and can be used as a local constitutive model on each integration point for concurrent structural-scale simulation of industrial components. However, their inherent simplifying hypotheses on the idealized microstructure may not be appropriate for other complex morphologies. In this case, full-field simulation of the micromechanical problem would be necessary, following a computational homogenization approach [6]. For instance, finite element (FE) simulation of a Representative Volume Element (RVE) of the underlying microstructure can be carried out to compute the local solution fields. The accurate linear and nonlinear effective responses can then be obtained through field averaging. However, such full-field method involves much higher computational cost and is impractical to be run concurrently in multiscale simulations on industrial parts.

Recently, the Deep Material Network (DMN) method [7] has emerged as a novel data-driven surrogate model for such multiscale materials. Based on linear elastic training data generated by high-fidelity computational homogenization (FE-RVE simulation, for instance), DMN is capable of predicting their elastic and inelastic behaviors with high accuracy and efficiency even in the finite-strain range. Compared to other data-driven approaches for material modeling [8] that rely on the (effective) strain-stress pair data, DMN learns instead the morphological characterizations of the microstructure via its homogenization function $(\mathbb{C}_1, \mathbb{C}_2) \mapsto \overline{\mathbb{C}}$. Here, $(\mathbb{C}_1, \mathbb{C}_2)$ are the elasticity tensors of the constituents and $\overline{\mathbb{C}}$ is the effective elasticity tensor. Due to its physics-based neural-network like architecture, DMN demonstrates great expressive power with much fewer fitting parameters compared to traditional machine learning methods. Thanks to these interesting properties, it is now gaining popularity in the computational mechanics community:

Email address: tianyi.li@3ds.com (Tianyi Li)

- In [9], DMN has been successfully applied for 3-d particle-reinforced composites, polycrystalline materials and woven composites involving three modeling scales. Nonlinear complex material behaviors such as hyperelasticity with stress softening and rate-dependent crystal plasticity are investigated under a finite-strain setting.

- A new rotation-free formulation along with a flattened architecture is proposed in [10]. It has been tested for short-fiber plastics and metal matrix composites with an elasto-plastic behavior in the matrix phase.

- Applications to short-fiber reinforced materials are also considered further in [11, 12, 13]. The thermomechanical behaviors of such composites can be successfully captured by DMN. Concurrent multiscale simulations combining a macroscopic finite element model and DMN on each integration point are performed for static and dynamic problems on structural components.

- More complex material behaviors such as viscoplasticity or failure are investigated in [14, 15, 16, 17]. New training strategies are also proposed to obtain more accurate nonlinear responses.

- In [18], other DMN architectures are explored with mean-field homogenization based building blocks. Applications to woven microstructures are considered. A generalization of the original DMN framework based on interactions between discrete material nodes is also proposed in [19, 20]. It has been successfully applied to porous microstructures with an elasto-plastic behavior.

Despite these successes indicated by numerical evidences, the expressive power of DMN remains to be fully understood from a theoretical perspective. In [10], its micromechanical justifications are provided in the context of generalized standard materials [21]. Thanks to the hierarchical architecture based on an appropriate laminate microstructure as the building block, DMN is thermodynamically consistent and verifies the micromechanical bounds on the effective linear and nonlinear behaviors. Actually, DMN can be identified as a multiple-rank laminate microstructure in homogenization theory [22, 23], in which each constituent of a laminate is also a laminate. Sequential laminates (also called coated laminates) are a subset among such hierarchical microstructures, where one of the constituents in the lamination process is always the same. It is well known that the Hashin-Shtrikman bounds [24] can be attained by such finite-rank sequential laminates. In the case of an effectively isotropic composite of two isotropic phases, such bounds can be realized by a rank-3 laminate in 2-d or a rank-6 laminate for 3-d elasticity [25]. As noted in [23, pg. 294], sequential laminates can approximate the homogenization function $(\mathbb{C}_1, \mathbb{C}_2) \mapsto \overline{\mathbb{C}}$ associated with a two-phase microstructure to the second order in $\mathbb{C}_1 - \mathbb{C}_2$. Since sequential laminates are a proper subset of multi-rank laminates, these results can be naturally extended to DMN and hence provide some preliminary theoretical justifications of its expressive power.

In general, we are often dealing with a class of microstructures that present similar morphologies. These microstructures are characterized by several parameters defining the geometrical shapes of each constituent inside the RVE. For instance, the volume fraction parameter vf describes the overall relative volume of each constituent. Other parameters also exist, depending on the exact morphological characterization of the microstructure. These micromechanical parameters may have an important influence on the effective properties. Hence, it would be beneficial that DMN also captures such effects with high accuracy and efficiency. In [26, 27], a transfer-learning based approach is proposed to construct a DMN database for such parametric microstructures. Training of several DMN instances is performed sequentially following a pre-determined path in the parametric space. Piecewise linear interpolation is then used to include the dependence of DMN fitting parameters on the microstructural parameters. In their work, they only focus on microstructures that can be characterized by vf and a second-order orientation tensor [28].

Meanwhile, a regression-based approach is proposed in [11] for short-fiber reinforced materials with a constant fiber volume fraction. The fitting parameters of one single DMN instance are assumed dependent on the varying (principal) fiber orientation tensor (a_1, a_2, a_3). The offline training is performed using a total loss function that aggregates multiple microstructures with different (a_1, a_2, a_3) values. The main objective of this paper is to extend this formulation to generic parametric microstructures, with varying volume fractions and possibly other geometrical parameters. In particular, we investigate of the use of Physics-Informed Neural Networks (PINN) to account for the dependence of DMN parameters on the microstructural ones. Following other PINN approaches [29, 30], additional micromechanics-based constraints are included to improve generalization capability of our parametric DMN architecture.

Another objective of this paper is to recast DMN in a multiphysics setting. In the literature, DMN has been used extensively as a surrogate of the possibly nonlinear *mechanical* behaviors of heterogeneous materials. After training, DMN is able to predict accurately the effective (incremental) stiffness tensor which relates the (current)

macroscopic strains and stresses. Other physical behaviors have not been explored except in [12], where a two-way coupled thermomechanical problem is considered. The temperature-dependent stress-strain behaviors as well as the mechanically-induced self-heating are predicted. However, effective multiphysics properties, such as thermal conductivity or the coefficient of thermal expansion, are not explicitly computed by DMN. In this paper, we demonstrate that such multiphysics properties can also be accurately predicted by DMN, even though it is trained using isothermal linear elastic mechanical data.

The paper is organized as follows. The original DMN formulation [7, 9] is first reviewed and then recast in a multiphysics setting in Sect. 2. Extensions to parametric microstructures are then considered in Sect. 3, in which we introduce a new PINN-based parametric DMN architecture. The proposed method is further evaluated on three parametric microstructures in Sect. 4. The obtained numerical results highlight the effects of the introduced physical constraints. Finally, conclusions from the current work are summarized in Sect. 5.

2. Deep Material Network

Instead of learning the macroscopic (nonlinear) strain-stress behaviors of a particular microstructure Ω, DMN learns its elastic linear homogenization function

$$(\mathbb{C}_1, \mathbb{C}_2) \mapsto \overline{\mathbb{C}}, \tag{1}$$

where $(\mathbb{C}_1, \mathbb{C}_2)$ are the elastic stiffness tensors of the constituents and $\overline{\mathbb{C}}$ is the homogenized stiffness tensor, see Fig. 1. The potentially anisotropic $(\mathbb{C}_1, \mathbb{C}_2)$ tensors are expressed in their respective material frames, while $\overline{\mathbb{C}}$ is given in the global frame of the microstructure Ω. The effective computation of (1) on complex microstructures Ω is in general realized by computational homogenization. The homogenization function encodes in particular the microstructure morphology, which can be defined by the characteristic function

$$\chi(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \text{Phase 1}, \\ 0 & \text{otherwise}. \end{cases} \tag{2}$$

In this regard, DMN is not a surrogate of the *effective behaviors* of the microstructure but the microstructure *per se*.

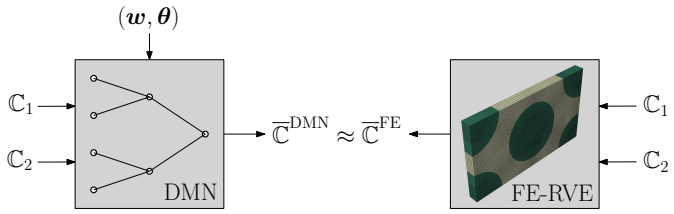


Figure 1: DMN learns the morphology of a particular microstructure Ω through its homogenization function, realized in general by computational homogenization like FE-RVE methods.

2.1. Network architecture and fitting parameters

In this work, we follow the original formulation of [7, 9] based on a perfect binary tree architecture and a rotation-based formulation for laminates, with minor modifications that will be explained in the sequel.

DMN is a multiple-rank laminate microstructure as defined in [23, Chap. 9], comprised of hierarchically nested laminates of laminates on different length scales, see Fig. 2. The rank n of such multiple-rank laminates, also called the number of DMN layers, characterizes the number of nesting levels. Its architecture corresponds topologically to a perfect binary tree. Each "node" is a rank-1 laminate microstructure, serving as the "mechanistic building blocks" [7, 9, 10] or "neurons" in this neural-network like architecture.

On the input layer, each leaf laminate of the binary tree receives the constitutive properties of the two phases $(\mathbb{C}_1, \mathbb{C}_2)$ expressed in their respective material frames. All these $\mathsf{n} = 2^n$ entries are called DMN material nodes, since each of them will be associated with a (nonlinear) constitutive behavior of the phase 1 or 2 during online prediction. The index set of the material nodes will be denoted $\mathbb{I}$. It can be partitioned evenly into two subsets for each of the phase.

$$\mathbb{I} = \{1, \dots, \mathsf{n}\} = \mathbb{I}_1 \cup \mathbb{I}_2, \quad \mathbb{I}_1 = \{2j-1 \mid j = 1, \dots, 2^{n-1}\}, \quad \mathbb{I}_2 = \{2j \mid j = 1, \dots, 2^{n-1}\}. \tag{3}$$

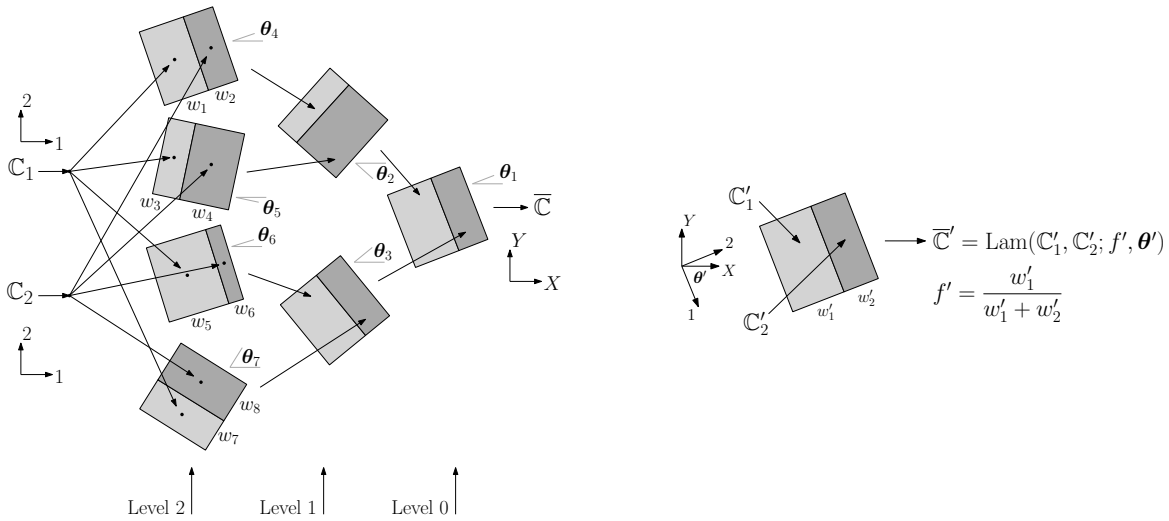


Figure 2: DMN architecture can be represented by a perfect binary tree with each "node" being a rank-1 laminate microstructure. The DMN material nodes $1 \leq i \leq \mathsf{n} = 2^n$ are represented by dots $\cdot$ on the input layer and receive the constitutive properties $(\mathbb{C}_1, \mathbb{C}_2)$ of the two phases. In this example with 3 DMN layers, three nesting levels is involved in order to compute the homogenization function $\mathrm{DMN}_3 : (\mathbb{C}_1, \mathbb{C}_2) \mapsto \overline{\mathbb{C}}$. The fitting parameters are the weights $\boldsymbol{w}$ defined on the leaf laminates and the rotations $\boldsymbol{\theta}$ on each laminate.

On the output layer, the effective properties of the microstructure $\overline{\mathbb{C}}$ are predicted through the root laminate and are given in the global frame of the microstructure Ω. The homogenization function $\mathrm{DMN}_n : (\mathbb{C}_1, \mathbb{C}_2) \mapsto \overline{\mathbb{C}}$ of such n-layer DMN can be defined recursively as follows

$$\overline{\mathbb{C}} = \mathrm{DMN}_n(\mathbb{C}_1, \mathbb{C}_2) = \begin{cases} \mathrm{Lam}(\mathbb{C}_1, \mathbb{C}_2) & n = 1, \\ \mathrm{Lam}\big(\mathrm{DMN}_{n-1}(\mathbb{C}_1, \mathbb{C}_2), \mathrm{DMN}_{n-1}(\mathbb{C}_1, \mathbb{C}_2)\big) & n > 1. \end{cases} \tag{4}$$

In (4), each Lam denotes the homogenization function of the rank-1 laminate associated with different weights (w_1', w_2') and rotations $\boldsymbol{\theta}'$, see Fig. 2. Compared to the original formulation [7], we omit the material rotation matrices for phase constitutive properties on the input layer. This is somehow similar to the formulation proposed by [10], even though the definition of the laminate homogenization function differs. Note that by omitting these input rotations, the material frames of the input properties will actually coincide with the local laminate frames.

A rank-1 laminate is a microstructure in which its characteristic function (2) only varies in the direction of lamination $\chi(\boldsymbol{x}) = \chi(\boldsymbol{x} \cdot \boldsymbol{n})$. In 3-d, its homogenization function $\mathrm{Lam} : (\mathbb{C}_1', \mathbb{C}_2') \mapsto \overline{\mathbb{C}}'$ can thus be characterized by only two parameters: the volume fraction f' of one of the phases, and a 3×3 rotation matrix $\boldsymbol{R}(\boldsymbol{\theta}') \in \mathrm{SO}(3)$ of the laminate with respect to *another* frame, see [7, 9]. The primes on the parameters and on the input and output tensors indicate that they don't represent necessarily the properties of the constituents (except for the input layer), nor the effective behaviors of the microstructure (except for the output layer). Thanks to the simplifying geometry, the homogenization function Lam is analytical and is explicitly defined in [7, 9]. For completeness, its definition is also recalled in Sect. 2.3.

In [10], the rotation matrix is replaced by the lamination normal vector $\boldsymbol{n}$, based on the lamination formulas in [23, Chap. 9]. This results in fewer fitting parameters since only two scalars $(n_1, n_2, \sqrt{1 - n_1 - n_2})$ are required to represent a unit-vector. The differences between these two formulations can be summarized as follows

- In [7, 9], the constituent properties $(\mathbb{C}_1', \mathbb{C}_2')$ are provided in the laminate frame $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$. The lamination direction $\boldsymbol{n}$ aligns with this local frame: $\boldsymbol{e}_2$ for 2-d microstructures, and $\boldsymbol{e}_3$ for 3-d ones. The homogenized behaviors of the laminate are first computed locally, then rotated to the frame of the next nesting level or the global frame for the laminate on level 0 (cf. Fig. 2). This will be referred to as the rotation-based formulation.
- In [10], the tensors $(\mathbb{C}_1', \mathbb{C}_2')$ and the lamination direction $\boldsymbol{n} \in \boldsymbol{R}^3$ are given directly with respect to the global frame. It is called the rotation-free formulation.

It is demonstrated in [10] that the rotation-free formulation also achieves satisfying expressivity for microstructures without local material orientation. Otherwise, for example when the inclusions are transversely isotropic with a spatially-varying orientation distribution, additional rotation matrices are required on the input layer, as mentioned by [10].

In summary, the fitting parameters of an n-layer DMN are

- Weights $\boldsymbol{w} = (w_1, \dots, w_{\mathsf{n}})$, defined on the leaf laminates. Its length $\mathsf{n} = 2^n$ is the number of DMN material nodes. They are required to be non-negative, hence an adequate nonlinear activation function σ is applied

$$\boldsymbol{w} = \sigma(\boldsymbol{z}) \geq 0, \quad \boldsymbol{z} \in \mathbb{R}^{\mathsf{n}}. \tag{5}$$

  ReLU $\sigma(z) = \max(0, z)$ is used in [7, 9, 10, 11, 18], while Softplus $\sigma(z) = \log(1+\exp(\beta z))/\beta$ is used in [19, 20] as a smooth approximation to ReLU.

  These weights are propagated from the leaf laminates to the root so that each laminate receives different (w'_1, w'_2), see Fig. 2. Denoting the weights on level $0 \leq i < n$ by $\boldsymbol{w}'^{(i)}$, with $\boldsymbol{w}'^{(n-1)} = \boldsymbol{w}$ the weights on the leaf laminates, we have

$$\boldsymbol{w}'^{(i)}_j = \boldsymbol{w}'^{(i+1)}_{2j-1} + \boldsymbol{w}'^{(i+1)}_{2j}, \quad j = 1, \dots, 2^{i+1}. \tag{6}$$

  For each laminate, the volume fraction f' can then be computed

$$f' = \frac{w'_1}{w'_1 + w'_2}. \tag{7}$$

  Due to (6) and (7), the DMN homogenization function (4) is invariant with respect to a scaling on DMN weights $\boldsymbol{w} \mapsto k\boldsymbol{w}$ for $k > 0$.

- Rotations $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{2^n-1})$, defined for each laminate. Its length $2^n - 1$ equals to the number of "nodes" in a perfect binary tree. Its elements can be organized using breadth-first ordering like in Fig. 2. Each element $\boldsymbol{\theta}_i$ represents a 3×3 rotation matrix $\boldsymbol{R}(\boldsymbol{\theta}_i)$. In [9, 18], the three Euler angles are used. In this work however, we use the quaternion representation which avoids singularity issues with Euler angles [31]. Hence, $\boldsymbol{\theta}$ is a $2^n - 1 \times 4$ matrix.

The total number of fitting parameters can be found in Tab. 1.

n-layer DMN	Weights	Rotations	Total
Number of fitting parameters	$\mathsf{n} = 2^n$	$4 \times (2^n - 1)$	$5 \times 2^n - 4$

Table 1: Number of fitting parameters for an n-layer DMN, using the network architecture explained in Sect. 2.1 with in particular the quaternion representation for the rotation matrices.

The training of these fitting parameters follows a data-driven approach, by comparing the DMN prediction of the homogenized elasticity tensor $\overline{\mathbb{C}}_{\text{DMN}}$ and that predicted by computational homogenization, see Fig. 1. The machine learning procedure is described in [7, 9, 10] and will be briefly recalled in Sect. 3.5 for our parametric DMN architecture.

2.2. Nonlinear homogenization formulation for online prediction

After training, DMN not only provides an accurate and efficient surrogate of the linear homogenization function (4), it serves also as a surrogate microstructure to predict its nonlinear behaviors. As for mean-field homogenization approaches, a linearization scheme is required [1] to linearize nonlinear constitutive laws. In [9], an incrementally affine formulation similar to [32] is proposed for DMN. In this work, we continue to use this nonlinear homogenization formulation, since it leads to a fixed-point problem that can be efficiently solved by acceleration methods [33]. In comparison, an incremental variational principle used in [10] results in a nonlinear system. Newton's method needs to be used, which requires constructing explicitly the Jacobian matrix and solving tangent linear systems.

Consider a time interval $[t_n, t_{n+1}]$ and an increment in the macroscopic strain from $\overline{\boldsymbol{\varepsilon}}_n$ to $\overline{\boldsymbol{\varepsilon}}_{n+1} = \overline{\boldsymbol{\varepsilon}}_n + \Delta\overline{\boldsymbol{\varepsilon}}$. As explained in Sect. 2.1, a material node is attached to each phase of the leaf laminates, see Fig. 2. At time t_n, we suppose that all their material state variables (strains, stresses and internal state variables) are known. The objective of the DMN surrogate is to compute at t_{n+1} the new homogenized stress tensor and the consistent tangent operator

$$\Delta\overline{\boldsymbol{\varepsilon}} \mapsto (\overline{\mathbb{C}}, \overline{\boldsymbol{\sigma}}_{n+1}). \tag{8}$$

The incrementally affine formulation assumes that for each material node, the stress increment in this time interval can be computed as follows

$$\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n = \Delta\boldsymbol{\sigma} = \mathbb{C}\Delta\boldsymbol{\varepsilon} + \delta\boldsymbol{\sigma}, \tag{9}$$

where $\Delta\boldsymbol{\varepsilon}$ is the strain increment for this material node, $\delta\boldsymbol{\sigma}$ the residual stress increment which measures material nonlinearity in this time interval and $\mathbb{C}$ is the consistent tangent operator. Note that in (9), the strain increment $\Delta\boldsymbol{\varepsilon}$ is also unknown, since only the increment in the macroscopic strain tensor $\Delta\overline{\boldsymbol{\varepsilon}}$ is known. A consequence of (9) is that the effective stress tensor increment can also be put in this incrementally affine form

$$\overline{\boldsymbol{\sigma}}_{n+1} - \overline{\boldsymbol{\sigma}}_n = \Delta\overline{\boldsymbol{\sigma}} = \overline{\mathbb{C}}\Delta\overline{\boldsymbol{\varepsilon}} + \overline{\delta\boldsymbol{\sigma}}. \tag{10}$$

The computation of (8) involves the following three *coupled* operations in the DMN architecture, see Fig. 3. They are explicitly defined in [9].

1. Forward homogenization: The consistent tangent operators $\mathbb{C}$ and residual stresses $\delta\boldsymbol{\sigma}$ defined on each material node are homogenized from the leaf laminates to the root laminate, obtaining the homogenized tensors $\overline{\mathbb{C}}_i$ and $\overline{\delta\boldsymbol{\sigma}}_i$ for each laminate
$$\text{Forward} : (\mathbb{C}_i, \delta\boldsymbol{\sigma}_i)_{1\le i\le \mathsf{n}} \mapsto (\overline{\mathbb{C}}_i, \overline{\delta\boldsymbol{\sigma}}_i). \tag{11}$$
2. Backward localization: The strain increment for each material node $\Delta\boldsymbol{\varepsilon}_i$ is computed based on the macroscopic strain increment $\Delta\overline{\boldsymbol{\varepsilon}}$ and the strain concentration tensors defined in each laminate obtained during forward homogenization, from the root laminate to the leaf laminates
$$\text{Backward} : (\overline{\mathbb{C}}_i, \overline{\delta\boldsymbol{\sigma}}_i; \Delta\overline{\boldsymbol{\varepsilon}}) \mapsto (\Delta\boldsymbol{\varepsilon}_i)_{1\le i\le \mathsf{n}}. \tag{12}$$
3. Constitutive behavior integration: For each material node $\mathbb{C}_i$ and $\delta\boldsymbol{\sigma}_i$ are computed by integrating constitutive behaviors, given their respective strain increment $\Delta\boldsymbol{\varepsilon}_i$.
$$\text{MatInt} : (\Delta\boldsymbol{\varepsilon}_i)_{1\le i\le \mathsf{n}} \mapsto (\mathbb{C}_i, \delta\boldsymbol{\sigma}_i)_{1\le i\le \mathsf{n}}. \tag{13}$$

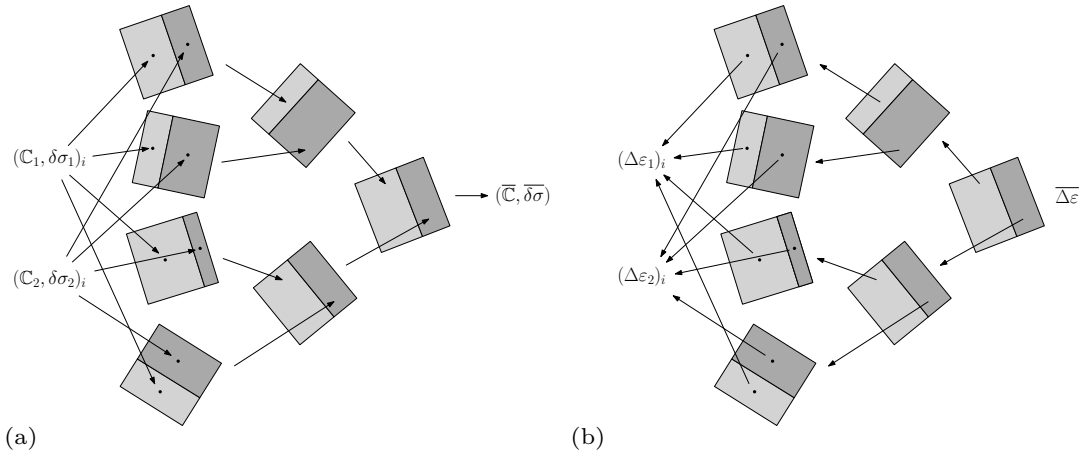


Figure 3: Forward homogenization and backward localization for online nonlinear behavior prediction of the microstructure using the DMN surrogate.

These coupled equations (11), (12) and (13) naturally give rise to a *vectorial* fixed point problem $\underline{x} = \mathsf{F}(\underline{x})$ on the flattened vector of strain increments $\underline{x} = (\Delta\boldsymbol{\varepsilon}_i)_{1\le i\le \mathsf{n}}$ for the time interval $[t_n, t_{n+1}]$

$$\underline{x} = (\underbrace{\text{Backward} \circ \text{Forward} \circ \text{MatInt}}_{\mathsf{F}})(\underline{x}) \implies \underline{x}^{j+1} = \mathsf{F}(\underline{x}^j) \tag{14}$$

where $j \ge 0$ is the fixed-point iteration number.

In this work, acceleration methods [33] will be used to accelerate the convergence of the fixed-point problem (14). In particular, Aitken relaxation [34, 35] is applied as a post-processing step in (14)

$$\begin{aligned} \underline{x}^{j+1} &= \mathsf{F}(\underline{x}^j) + (\omega^j - 1)\underline{r}^j, \quad \underline{r}^j = \mathsf{F}(\underline{x}^j) - \underline{x}^j \\ \omega^j &= \begin{cases} \omega^0 & j = 0 \\ -\omega^{j-1}\dfrac{(\underline{r}^{j-1})^{\mathsf{T}}(\underline{r}^j - \underline{r}^{j-1})}{\|\underline{r}^j - \underline{r}^{j-1}\|^2} & j \ge 1 \end{cases} \end{aligned} \tag{15}$$

In (15), after updating the relaxation parameter ω^j, it is then numerically bounded by $[\omega_{\min}, \omega_{\max}]$. Its initial value ω^0 as well as the bounds will be discussed in Sect. 4. The algorithm that realizes nonlinear homogenization (8) in $[t_n, t_{n+1}]$ is summarized as follows.

- Given macroscopic strain increment $\Delta\overline{\boldsymbol{\varepsilon}}$ with $\|\Delta\overline{\boldsymbol{\varepsilon}}\| > \epsilon$. In this work we use $\epsilon = 10^{-8}$.
- Initialize the strain increments $\underline{x}^0$ on material nodes, by applying backward localization (12) with the strain concentration tensors computed at t_n. For the first time increment t_0, each material node and laminate is initialized using linear elastic properties.
- For the fixed-point iteration $j \geq 0$:
  1. Integrate constitutive behaviors for each material node (13), obtaining $(\mathbb{C}_i, \delta\boldsymbol{\sigma}_i)_{1\leq i\leq \mathtt{n}}$.
  2. Forward homogenization (11), obtaining $(\overline{\mathbb{C}}_i, \overline{\delta\boldsymbol{\sigma}}_i)$.
  3. Backward localization (12), obtaining $\mathsf{F}(\underline{x}^j)$.
  4. Check convergence. In this work, the relative error in the L_2 norm with respect to the macroscopic strain increment is controlled
  $$e_{\text{rel}} = \frac{\|\mathsf{F}(\underline{x}^j) - \underline{x}^j\|}{\|\Delta\overline{\boldsymbol{\varepsilon}}\|} < \text{tol}. \tag{16}$$
  If convergence is reached, end the fixed-point iterations.
  5. Apply Aitken relaxation (15), obtaining $\underline{x}^{j+1}$.
- Compute the effective stress tensor (10).

If all the material nodes remain linear elastic, only 1 fixed-point iteration is required, since $\|\mathsf{F}(\underline{x}^0) - \underline{x}^0\| = 0$ in (16).

2.3. Multiphysics property prediction

Up to now, we have only considered the possibly nonlinear mechanical behaviors of the heterogeneous material. As noted at the beginning of Sect. 2 and in Fig. 1, DMN should not be regarded as a surrogate of the *effective mechanical behaviors* but the microstructure *per se*. It is hence desirable that other physical properties can be predicted by DMN along with the mechanical ones at the online prediction stage. In this paper, we propose to consider the following two additional physical properties

1. Thermal conductivity $\boldsymbol{k}$, which is a symmetric second-order tensor for anisotropic behaviors.
2. Coefficient of thermal expansion $\boldsymbol{\alpha}$, also a symmetric second-order tensor in the anisotropic case.

It turns out that the online prediction of $\boldsymbol{k}$ and $\boldsymbol{\alpha}$ requires only an adequate redefinition of the building block (neuron), denoted by Lam in (4). In particular, the DMN network architecture as well as the fitted parameters after offline training remain exactly the same. This property naturally recasts DMN in a multiphysics setting. Although each DMN neuron corresponds to a laminate microstructure, the exact formula of Lam will now depend on the physics being considered. Below, we will provide the definition of Lam for these additional physics properties. For completeness, its original definition [7, 9] for (linear elastic) mechanical behaviors is also included.

To ease the notation, we will now ignore the primes on the input and output tensors in the laminate homogenization function, see Fig. 1. For instance, for linear elasticity we now have

$$\overline{\mathbb{C}} = \text{Lam}(\mathbb{C}_1, \mathbb{C}_2).$$

It should be remembered that the input tensors don't necessarily refer to those of the two constituents (except for the input layer), and the output effective tensor is not necessarily the final effective property tensor of the microstructure (except for the output layer).

Linear elastic behaviors

The linear elastic behaviors of the laminate microstructure is governed by static equilibrium. Based on the lamination direction $\boldsymbol{n}$, the stress components can be partitioned into two parts: a tangential part $\boldsymbol{\sigma}^{\text{t}}$ and a normal part $\boldsymbol{\sigma}^{\text{n}}$. For example, if the lamination direction aligns with the local $\boldsymbol{e}_3$ vector, then we have

$$\boldsymbol{\sigma}^{\text{t}} = (\sigma_{11}, \sigma_{22}, \sigma_{12}), \quad \boldsymbol{\sigma}^{\text{n}} = (\sigma_{33}, \sigma_{13}, \sigma_{23}). \tag{17}$$

The strain tensor as well as the stiffness tensors can be partitioned similarly. The interface condition prescribes that the normal part of the stress tensor $\boldsymbol{\sigma}^{\text{n}}$ is continuous, while the tangential part of the strain tensor $\boldsymbol{\varepsilon}^{\text{t}}$ is continuous. It can be encoded by the following matrix equation

$$\begin{bmatrix} \boldsymbol{\varepsilon}_1^{\text{t}} \\ \boldsymbol{\sigma}_1^{\text{n}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varepsilon}_2^{\text{t}} \\ \boldsymbol{\sigma}_2^{\text{n}} \end{bmatrix} \implies \underbrace{\begin{bmatrix} \mathbb{I} & 0 \\ \mathbb{C}_1^{\text{nt}} & \mathbb{C}_1^{\text{nn}} \end{bmatrix}}_{\widehat{\mathbb{C}}_1} \begin{bmatrix} \boldsymbol{\varepsilon}_1^{\text{t}} \\ \boldsymbol{\varepsilon}_1^{\text{n}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbb{I} & 0 \\ \mathbb{C}_2^{\text{nt}} & \mathbb{C}_2^{\text{nn}} \end{bmatrix}}_{\widehat{\mathbb{C}}_2} \begin{bmatrix} \boldsymbol{\varepsilon}_2^{\text{t}} \\ \boldsymbol{\varepsilon}_2^{\text{n}} \end{bmatrix}, \tag{18}$$

where $\mathbb{C}_1$ and $\mathbb{C}_2$ are the stiffness tensors of the two laminate phases.

Using the definition of the effective strain tensor $\overline{\boldsymbol{\varepsilon}} = f\boldsymbol{\varepsilon}_1 + (1-f)\boldsymbol{\varepsilon}_2$, where f is the volume fraction of the phase 1, we obtain the following formula that computes $\boldsymbol{\varepsilon}_1$ from $\overline{\boldsymbol{\varepsilon}}$

$$\big((1-f)\widehat{\mathbb{C}}_1 + f\widehat{\mathbb{C}}_2\big)\boldsymbol{\varepsilon}_1 = \widehat{\mathbb{C}}_2\overline{\boldsymbol{\varepsilon}} \implies \boldsymbol{\varepsilon}_1 = \mathbb{A}\overline{\boldsymbol{\varepsilon}} = \widehat{\mathbb{C}}^{-1}\widehat{\mathbb{C}}_2\overline{\boldsymbol{\varepsilon}}, \quad \widehat{\mathbb{C}} = (1-f)\widehat{\mathbb{C}}_1 + f\widehat{\mathbb{C}}_2, \tag{19}$$

where $\mathbb{A} = \widehat{\mathbb{C}}^{-1}\widehat{\mathbb{C}}_2$ is the strain localization tensor.

Finally, with the effective stress tensor $\overline{\boldsymbol{\sigma}} = f\boldsymbol{\sigma}_1 + (1-f)\boldsymbol{\sigma}_2 = f\mathbb{C}_1\boldsymbol{\varepsilon}_1 + (1-f)\mathbb{C}_2\boldsymbol{\varepsilon}_2$, the effective stiffness tensor can be obtained as follows

$$\overline{\boldsymbol{\sigma}} = \overline{\mathbb{C}}\overline{\boldsymbol{\varepsilon}}, \quad \overline{\mathbb{C}} = f(\mathbb{C}_1 - \mathbb{C}_2)\mathbb{A} + \mathbb{C}_2. \tag{20}$$

The rotation can then be applied adequately for the fourth-order stiffness tensor $\overline{\mathbb{C}}$.

Thermal conductivity

Now, the neuron homogenization function computes the second-order effective (anisotropic) conductivity tensor $\overline{\boldsymbol{k}}$ from those of its constituents $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$

$$\overline{\boldsymbol{k}} = \mathrm{Lam}(\boldsymbol{k}_1, \boldsymbol{k}_2).$$

The laminate microstructure is governed by steady-state heat equation and the Fourier's law $\boldsymbol{q} = -\boldsymbol{k}\nabla T$ that relates the temperature gradient ∇T to the heat flux $\boldsymbol{q}$ through the conductivity tensor. As in (17), the heat flux and the temperature gradient can be decomposed into a tangential and a normal part

$$\boldsymbol{q}^{\mathrm{t}} = (q_1, q_2), \quad \boldsymbol{q}^{\mathrm{n}} = (q_3).$$

Similar to (18), the interface condition can be described by the following linear system

$$\begin{bmatrix} \nabla T_1^{\mathrm{t}} \\ \boldsymbol{q}_1^{\mathrm{n}} \end{bmatrix} = \begin{bmatrix} \nabla T_2^{\mathrm{t}} \\ \boldsymbol{q}_2^{\mathrm{n}} \end{bmatrix} \implies \underbrace{\begin{bmatrix} \mathbb{I} & 0 \\ \boldsymbol{k}_1^{\mathrm{nt}} & \boldsymbol{k}_1^{\mathrm{nn}} \end{bmatrix}}_{\widehat{\boldsymbol{k}}_1} \begin{bmatrix} \nabla T_1^{\mathrm{t}} \\ \nabla T_1^{\mathrm{n}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbb{I} & 0 \\ \boldsymbol{k}_2^{\mathrm{nt}} & \boldsymbol{k}_2^{\mathrm{nn}} \end{bmatrix}}_{\widehat{\boldsymbol{k}}_2} \begin{bmatrix} \nabla T_2^{\mathrm{t}} \\ \nabla T_2^{\mathrm{n}} \end{bmatrix}.$$

Following the same procedure as before, the effective conductivity tensor is given by

$$\overline{\boldsymbol{q}} = \overline{\boldsymbol{k}}\overline{\nabla T}, \quad \overline{\boldsymbol{k}} = f(\boldsymbol{k}_1 - \boldsymbol{k}_2)\mathbf{A} + \boldsymbol{k}_2, \quad \mathbf{A} = \widehat{\boldsymbol{k}}^{-1}\widehat{\boldsymbol{k}}_2, \quad \widehat{\boldsymbol{k}} = (1-f)\widehat{\boldsymbol{k}}_1 + f\widehat{\boldsymbol{k}}_2. \tag{21}$$

The rotation can be applied to the second-order tensor $\overline{\boldsymbol{k}}$.

Coefficient of thermal expansion

The effective coefficient of thermal expansion (CTE), as a second-order tensor in the anisotropic case, can be computed along with the effective stiffness tensor. Now, the laminate homogenization function becomes

$$(\overline{\mathbb{C}}, \overline{\boldsymbol{\alpha}}) = \mathrm{Lam}(\mathbb{C}_1, \mathbb{C}_2, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2)$$

While $\overline{\mathbb{C}}$ can be obtained by (20), the effective CTE is computed using the following relation [36]

$$\overline{\boldsymbol{\alpha}} = \boldsymbol{\alpha}_1 + (\mathbb{S}_1 - \overline{\mathbb{S}})(\mathbb{S}_2 - \mathbb{S}_1)^{-1}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2), \tag{22}$$

where $\mathbb{S} = \mathbb{C}^{-1}$ is the compliance tensor.

Note that (22) should also be used as a *neuron* operation similar to (20) and (21), layer by layer. All the input and output tensors need to be expressed and computed in the same laminate frame $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$. This is particularly true when local material orientation is present for one of the constituents. In such case, the involved stiffness and CTE tensors will be expressed in the local material frame that varies spatially in the microstructure. Meanwhile, if local material orientation is absent, *i.e.* the input constituent as well as the effective properties are expressed in the global frame, then (22) can also be applied directly using the effective compliance tensor $\overline{\mathbb{S}}$ computed by DMN.

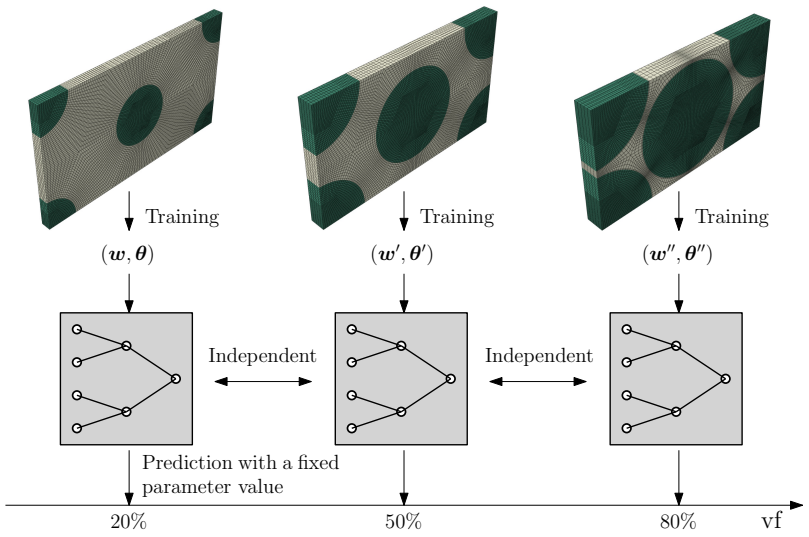


Figure 4: With the original formulation, each DMN is trained independently to learn a particular microstructure $\Omega_{\boldsymbol{p}_i}$. It is not possible to predict the behavior of a new microstructure $\Omega_{\boldsymbol{p}'}$ without another training.

3. Extension to parametric microstructures

In general, we are often dealing with a class of microstructures which present similar morphologies. These microstructures are defined by a characteristic function involving one or several parameters $\boldsymbol{p} = (p_1, p_2, \ldots)$. In the case of hex-packed unidirectional fiber composites, illustrated in Fig. 4, they are parametrized by the single vf parameter which indicates the volume fraction of the fibers.

The original DMN formulation [7] is not adapted for such parametric microstructures $\boldsymbol{p} \mapsto \Omega_{\boldsymbol{p}}$, since each DMN is trained to learn a particular microstructure $\Omega_{\boldsymbol{p}_i}$ with a fixed parameter value $\boldsymbol{p}_i$. After independent training on such n microstructures, we only obtain a *discrete* set of DMN objects

$$\{\mathrm{DMN}_{(\boldsymbol{w}_i, \boldsymbol{\theta}_i)}, \quad i = 1, \ldots, n\}.$$

Each $\mathrm{DMN}(\boldsymbol{w}_i, \boldsymbol{\theta}_i)$ is tailored to approximate the homogenized behavior of a *particular* microstructure $\Omega_{\boldsymbol{p}_i}$. Without another training (which requires the costly computational homogenization data), it is not possible to predict the behavior of a new microstructure $\Omega_{\boldsymbol{p}'}$ that is absent in the previous set.

3.1. Transfer-learning based interpolative DMN

In [26], a transfer-learning strategy is proposed to interpolate different DMN models trained at different parameter values $\boldsymbol{p}_i$. The functional dependence $\boldsymbol{p} \to \big(\boldsymbol{w}(\boldsymbol{p}), \boldsymbol{\theta}(\boldsymbol{p})\big)$ is defined by interpolating the DMN parameters $(\boldsymbol{w}_i, \boldsymbol{\theta}_i)$ between different microstructural parameters $\boldsymbol{p}_i$. This assumes implicitly that the same DMN architecture with the same number of layers is used. If each DMN is trained independently using random initialization of its fitting parameters, $\boldsymbol{p} \to \big(\boldsymbol{w}(\boldsymbol{p}), \boldsymbol{\theta}(\boldsymbol{p})\big)$ would not be *smooth*, which leads to less accurate predictions when interpolating between known $\boldsymbol{p}_i$. This motivates hence a transfer-learning based training strategy, illustrated in Fig. 5.

A *pre-determined sequential* path $(\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n)$ in the parametric space is required to carry out transfer-learning training on DMN parameters

$$\mathrm{DMN}_{(\boldsymbol{w}_0, \boldsymbol{\theta}_0)} \to \mathrm{DMN}_{(\boldsymbol{w}_1, \boldsymbol{\theta}_1)} \to \cdots \to \mathrm{DMN}_{(\boldsymbol{w}_n, \boldsymbol{\theta}_n)}\,.$$

The training of $(\boldsymbol{w}_i, \boldsymbol{\theta}_i)$ is initialized using previously trained $(\boldsymbol{w}_{i-1}, \boldsymbol{\theta}_{i-1})$, for $i > 0$. Only the first training at $\boldsymbol{p}_0$ is performed using standard random initialization. Transfer learning not only ensures a smooth interpolation between DMN parameters, it also accelerates training of subsequent DMN models for new microstructural parameters $\boldsymbol{p}_i$.

While this transfer-learning based approach has been tested with success for microstructures depending solely on the volume fraction parameter [26], and those depending on the volume fraction and the orientation tensor [27]. In our opinion it has also some limitations:

- Definition of a *pre-determined training sequence* may become difficult especially for higher parametric dimensions. For microstructures with multiple geometrical parameters of different natures, it is not trivial to choose

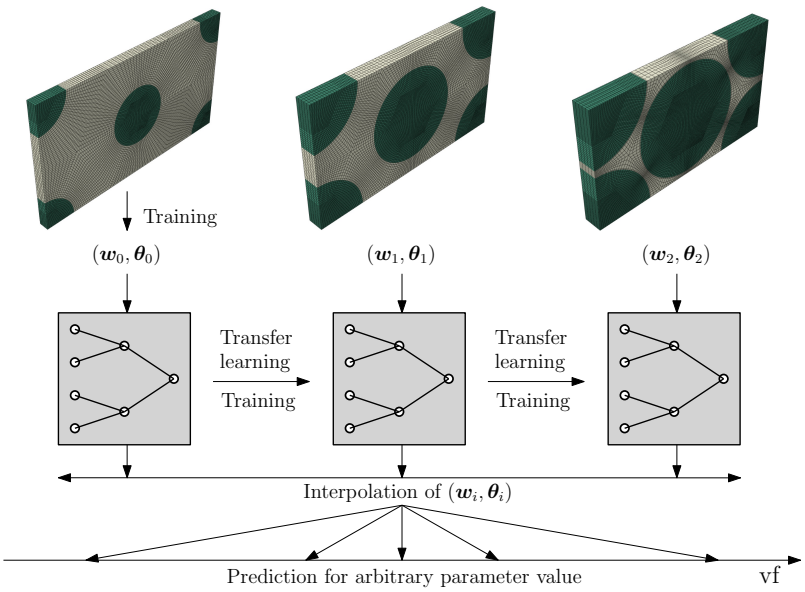


Figure 5: Transfer-learning based interpolative deep material network.

the starting point $\boldsymbol{p}_0$ as well as the sequential interpolation points $\boldsymbol{p}_i$. Furthermore, the expressive power of DMN in the parametric space may also depend on the chosen training path $(\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_n)$. The number of DMN active material nodes (those with a positive weight) may decrease in the process of transfer learning with the ReLU activation function. An example is given in [26].

- Extrapolation especially for higher parametric dimensions. Based on interpolation points, it would be difficult to extrapolate "outside" the training domain. In [26, 27] for instance, extrapolation in the parametric space is not considered.

- Computational efficiency concerns. Since transfer learning is sequential in nature, DMN at different microstructural parameters can only be trained one after another. For interpolation, each DMN instance associated with its parameters needs to be stored.

3.2. Physics-informed neural network-based parametric DMN

In this work, we propose a novel parametric DMN architecture dedicated for microstructures with parameters. The functional dependence of DMN parameters $(\boldsymbol{w}, \boldsymbol{\theta})$ on the microstructural parameters $\boldsymbol{p}$ is directly accounted for by feedforward neural networks composed of multiple layers of affine transformations and activation functions. Due to the physics-based architecture of DMN and its high expressive power of approximating arbitrary microstructures, hidden layers are ignored in this work. We obtain thus

$$\boldsymbol{w}(\boldsymbol{p}) = \sigma(\boldsymbol{W}_1 \boldsymbol{p} + \boldsymbol{w}_0), \tag{23a}$$

$$\boldsymbol{\theta}(\boldsymbol{p}) = \boldsymbol{\Theta}_1 \boldsymbol{p} + \boldsymbol{\theta}_0, \tag{23b}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{\Theta}_1$ are matrices of appropriate dimensions, $\boldsymbol{w}_0$ and $\boldsymbol{\theta}_0$ are "bias" vectors and σ denotes a nonlinear activation function as in (5). Several remarks are in order:

- The matrices $\boldsymbol{W}_1$ and $\boldsymbol{\Theta}_1$ characterize the dependence of DMN parameters $\boldsymbol{w}$ and $\boldsymbol{\theta}$ on microstructural parameters, while $\boldsymbol{w}_0$ and $\boldsymbol{\theta}_0$ remain constant while microstructural parameters change.

- When the microstructural parameters have little effect on the effective behaviors of the microstructures, we would have $\boldsymbol{W}_1 = \boldsymbol{\Theta}_1 = \boldsymbol{0}$ and thus recover the original DMN formulation [7]. The "bias" parameters $\boldsymbol{w}_0$ and $\boldsymbol{\theta}_0$ solely are needed to approximate the homogenized behavior of this parametric microstructure.

- Due to the presence of a nonlinear activation function applied after affine transformation, the functional dependence of DMN weights (23a) is different from piecewise linear interpolation used in the transfer-learning based interpolative DMN approach [26, 27].

- Similar functional dependence of DMN rotations (23b) has been proposed in [11].

The neural network proposed in (23) follows a *fully-connected* architecture in the sense that the DMN weights $\boldsymbol{w}$ and rotations $\boldsymbol{\theta}$ depend on *all* the microstructural parameters. Meanwhile, several findings in the literature motivate a physics-informed architecture:

- For parametric microstructures with a single *volume fraction* parameter, the naïve approach, as proposed by [26] in which only the DMN weights vary with vf and the DMN rotations remain constant, actually predicts physically plausible vf-dependence of the homogenized behaviors.
- For microstructures with parameters that do not change the volume fraction, such as for short-fiber reinforced composites with a fixed volume fraction but different fiber orientations, [11] indicates that only the DMN rotations need to vary with such parameters, with a constant DMN weights vector.

It should be noted that these results originate directly from the micromechanics-based design of DMN [10]. In order that the parametric DMN reflects these physical properties, the microstructural parameters are partitioned

$$\boldsymbol{p} = (\mathrm{vf}, \boldsymbol{q}) \in \mathbb{R} \times \mathbb{R}^q, \tag{24}$$

where vf is the volume fraction of the phase 2, while $\boldsymbol{q}$ denotes all other q *independent* parameters that are *orthogonal to* vf. Given the partition (24), the physics-informed neural network (PINN) for DMN parameters is now given by

$$\boldsymbol{w}(\boldsymbol{p}) = \boldsymbol{w}(\mathrm{vf}) = \sigma(\mathrm{vf} \cdot \boldsymbol{w}_1 + \boldsymbol{w}_0), \tag{25a}$$
$$\boldsymbol{\theta}(\boldsymbol{p}) = \boldsymbol{\theta}(\boldsymbol{q}) = \boldsymbol{\Theta}_1 \boldsymbol{q} + \boldsymbol{\theta}_0, \tag{25b}$$

where $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$ are vectors of length n, $\boldsymbol{\theta}_0$ is a $(2^n - 1) \times 4$ matrix and $\boldsymbol{\Theta}_1$ is a $(2^n - 1) \times 4 \times q$ tensor. The fully-connected architecture (23) and the PINN architecture (25) are compared in Fig. 6.

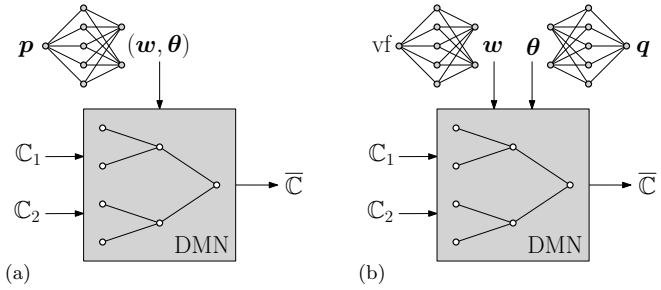


Figure 6: Parametric deep material network: (a) fully-connected architecture and (b) physics-informed architecture.

The advantages of the PINN architecture (25) compared to the fully-connected one is two-fold. Firstly, fewer fitting parameters are required due to the partition (24), which increases computational efficiency. Secondly, as we shall see through numerical examples, it provides comparable expressive power and may also enhance generalization ability (interpolation and extrapolation) in the parametric space.

Compared to the transfer learning-based interpolative DMN described in Sect. 3.1, a *unique* offline training is now required to optimize the fitting parameters of PINN-DMN, see Fig. 7. The expressive power of DMN is evaluated *jointly* using the linear elastic behavior data at each $\boldsymbol{p}_i$ in the parametric space. Furthermore, a neural-network functional dependence naturally defines interpolation and extrapolation inside or outside the training domain and extends easily to higher parametric dimensions.

This physics-informed neural network-based parametric DMN will be denoted in the sequel by PINN-DMN. To summarize, it is defined by

- DMN homogenization function (4) which learns the morphology of a particular microstructure $\Omega_{\boldsymbol{p}}$.
- Physics-informed neural network (25) which learns the functional dependence of DMN parameters on microstructural parameters $\boldsymbol{p}$. It contains the fitting parameters of PINN-DMN: $(\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{\theta}_0, \boldsymbol{\Theta}_1)$.

With an n-layer architecture, the number of fitting parameters can be found in Tab. 2. The offline training of these fitting parameters follows a data-driven machine learning approach which will be described in Sect. 3.5.

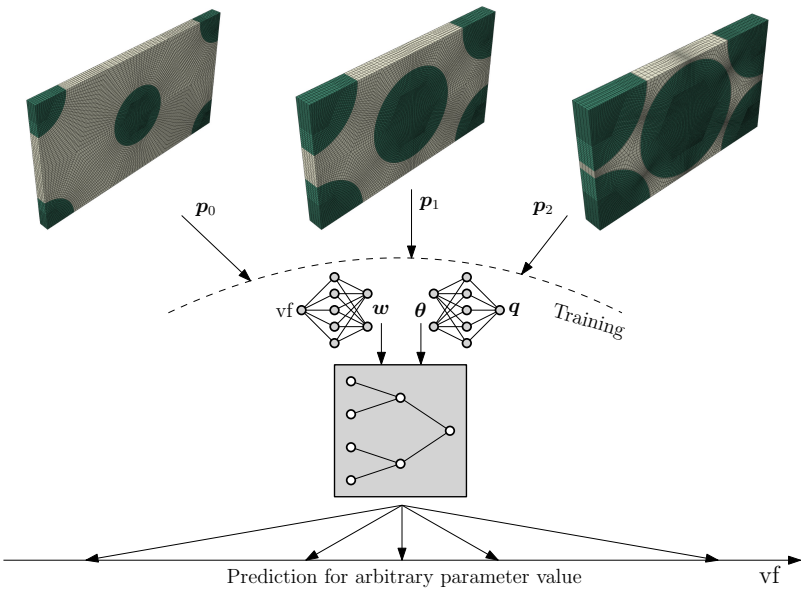


Figure 7: Offline training for PINN-DMN based on the linear elastic behavior data at each $\boldsymbol{p}_i$ in the parametric space.

n-layer PINN-DMN	Weights	Rotations
Number of fitting parameters	$2\mathsf{n} = 2^{n+1}$	$4(q+1) \times (2^n - 1)$

Table 2: Fitting parameters and their numbers for an n-layer PINN-DMN.

3.3. Volume fraction constraint

Apart from the physics-informed architecture on the functional dependence $\boldsymbol{p} \mapsto (\boldsymbol{w}, \boldsymbol{\theta})$, some physical constraints can also be prescribed to further improve its generalization ability, similar to the physics-informed machine learning approach for nonlinear partial differential equations [29, 30].

It is now well known that the DMN weights vector $\boldsymbol{w}$ reflects the actual volume fraction information of the microstructure Ω being considered [7, 10]. Specifically, the volume fraction of the phase 2 learned by DMN is given by

$$\mathsf{vf}(\boldsymbol{w}) = \frac{\sum_{i \in \mathbb{I}_2} w_i}{\sum_{i \in \mathbb{I}} w_i} \approx \mathrm{vf}_\Omega. \tag{26}$$

In [16], this physical property is explicitly included in the loss function as an additional constraint in learning a particular microstructure. Using our PINN functional dependence (25a), (26) can be further enforced at *all volume fraction* values

$$\mathsf{vf}(\boldsymbol{w}) = \mathsf{vf}\big(\sigma(\mathrm{vf} \cdot \boldsymbol{w}_1 + \boldsymbol{w}_0)\big) = \mathrm{vf}, \quad \forall \mathrm{vf} \in [0, 1]. \tag{27}$$

This volume fraction constraint is prescribed on DMN weights $\boldsymbol{w}$.

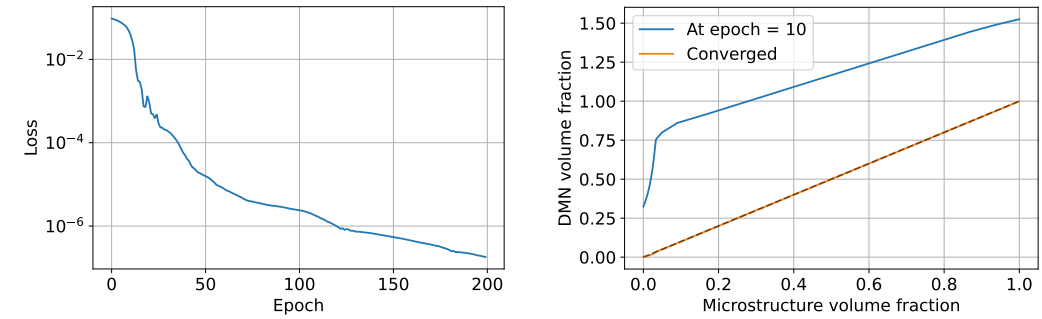
Due to the presence of a nonlinear activation function σ, (27) is nonlinear and the unknowns $(\boldsymbol{w}_0, \boldsymbol{w}_1)$ can not be solved explicitly. Furthermore, and luckily enough since it contributes to the expressive power of PINN-DMN, there may exist multiple solutions that satisfy (27). Anticipating the fact that this constraint will also be included during the offline training of our PINN-DMN, we adopt a machine-learning approach to weakly enforce this volume fraction constraint through a loss function

$$\mathcal{L}_{\mathrm{vf}} = \frac{1}{n} \sum_{i=1}^{n} \big(\mathsf{vf}\big(\sigma(\mathrm{vf}_i \cdot \boldsymbol{w}_1 + \boldsymbol{w}_0)\big) - \mathrm{vf}_i\big)^2, \tag{28}$$

where vf_i denotes the collocation points at which (27) is weakly prescribed and n is the number of such collocation points. The sampling of these collocation points, the initialization of $(\boldsymbol{w}_0, \boldsymbol{w}_1)$ and the optimization algorithm will be described together with the offline training of PINN-DMN in Sect. 3.5.

Using 5 DMN layers and the ReLU activation function, an example of the minimization of (28) is shown in Fig. 8. The loss function is decreasing, which demonstrates that our PINN functional dependence (25a) of $\boldsymbol{w}$ is capable

of satisfying the volume fraction constraint (27). In this example, after about 200 epochs, the maximum absolute error of $|\mathsf{vf}(\boldsymbol{w}) - \mathrm{vf}|$ in $[0, 1]$ becomes approximately 0.5%.



(a) (b)

Figure 8: Volume fraction constraint prescribed on the PINN for $\boldsymbol{w}(\boldsymbol{p})$: (a) minimization of the loss function (28) and (b) predicted DMN volume fraction at the early stage and at the end of training.

Another physical information that can be extracted from (25a) is the number of active DMN material nodes. For arbitrary $\mathrm{vf} \in [0, 1]$, the number of active nodes for the phase 1 is given by

$$\mathrm{card}\{i \in \mathbb{I}_1 \mid w_i(\mathrm{vf}) > 0\},$$

where $\mathrm{card}(\cdot) = |\cdot|$ is the cardinality of a set. Similar formula can be defined for the phase 2. In practice, machine epsilon is used instead of 0. The ratio of active DMN nodes for each phase can be obtained by dividing their respective number of active nodes by $\frac{1}{2}\mathsf{n}$, which is the number of total material nodes per phase. In the whole parametric space, the number of *globally* active nodes is defined by

$$\mathrm{card}\{i \in \mathbb{I} \mid w_i(\mathrm{vf}) > 0, \quad \forall \mathrm{vf} \in [0, 1]\}.$$

In Fig. 9, these quantities are plotted as a function of vf, the volume fraction of the phase 2. In order to satisfy (27), the ratio of active nodes of the phase 1 decreases from 100% to less than 70%, while for the phase 2 this ratio increases from less than 60% up to more than 90%. Our PINN functional dependence (25a) of $\boldsymbol{w}$ is hence capable of adapting the weights of each phase as a function of the microstructure volume fraction.

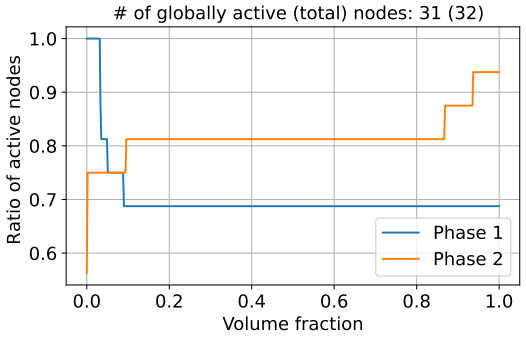


Figure 9: Variation of the ratio of active DMN nodes with the volume fraction of the phase 2.

3.4. Orientation constraint

DMN not only captures the microstructure morphology, it also learns material orientation distribution function $\Omega \to \mathrm{SO}(3)$ in the microstructure [9]. Using orientation tensors, we propose an orientation constraint to be prescribed on DMN weights $\boldsymbol{w}$ and rotations $\boldsymbol{\theta}$, in order that the parametric DMN generalizes such material orientation knowledge in the whole parametric space.

Orientation tensors [28] describe concisely the statistical information of the orientation distribution of *unit vectors*. Given an orientation distribution function $f : \mathbb{S}^2 \to \mathbb{R}$, where $\mathbb{S}^2$ denotes the two-dimensional surface of a unit sphere, the 3×3 second-order orientation tensor is defined by

$$\boldsymbol{a} = \int_{\mathbb{S}^2} f(\boldsymbol{e}) \boldsymbol{e} \otimes \boldsymbol{e} \, \mathrm{d}\boldsymbol{e}, \quad \|\boldsymbol{e}\| = 1. \tag{29}$$

It can be easily shown that $\boldsymbol{a}$ is symmetric and $\mathrm{tr}(\boldsymbol{a}) = 1$, due to the normalization constraints of $\boldsymbol{e}$ and of the probability density f. Higher order orientation tensors do exist [37], however the second-order one (29) is the most frequently used to characterize local fiber orientations due to manufacturing processes and their influence on material properties [38, 39].

In this work, we propose a generalization of such (second-order) orientation tensors for orientation distributions of *rotations*. Contrary to transversely isotropic fibers for which a single unit vector suffices to characterize its material frame, general anisotropic materials (like orthotropic ones) require a rotation matrix $\boldsymbol{R}$ to describe the transformation from its material frame $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ to the global one. In such cases, the orientation distribution function is now defined for rotation matrices $f : \mathrm{SO}(3) \to \mathbb{R}$. Such probability density function can also be found for the texture analysis of polycrystalline materials [40] or rigid body dynamics [41]. Given that each column $1 \leq i \leq 3$ of $\boldsymbol{R}$ essentially expresses $\boldsymbol{e}_i \in \mathbb{S}^2$ in the global frame, it defines an orientation distribution of the material frame, cf. Fig. 10 for the tows.

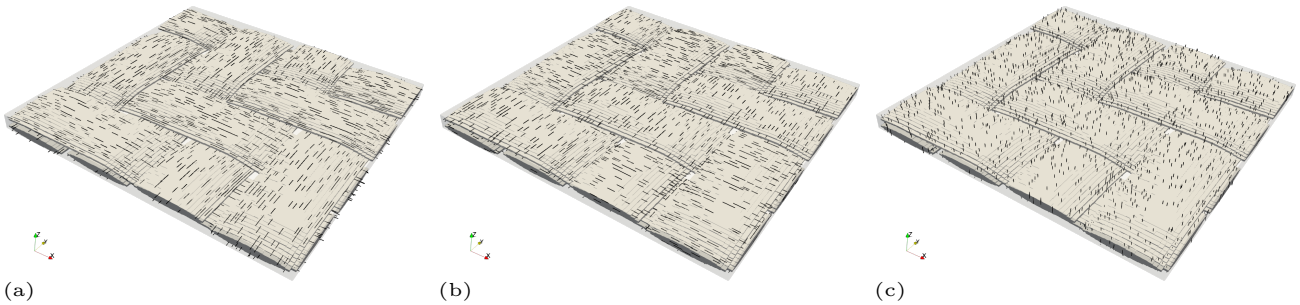


Figure 10: Orientation distribution of the material frame of the tows in the woven microstructure: (a) $\boldsymbol{e}_1$, (b) $\boldsymbol{e}_2$ and (c) $\boldsymbol{e}_3$.

Given this interpretation, three (second-order) orientation tensors can be defined for each axis of the material frame

$$\mathsf{a}^{(i)} = \int_{\mathbb{S}^2} f(\boldsymbol{R}) \boldsymbol{e}_i \otimes \boldsymbol{e}_i \, \mathrm{d}\boldsymbol{e}, \quad \|\boldsymbol{e}_i\| = 1, \tag{30}$$

where Einstein summation is not implied. The tensor a can be understood as a third-order material frame orientation tensor. For discrete probability functions defined on a mesh, the integral can be understood as weighted-averaging using the element volumes as weights. Note that in the case of a two-phase microstructure, (30) can be computed for each of the phase. Each of these orientation tensors satisfies $\mathrm{tr}(\mathsf{a}^{(i)}) = 1$. Due to the orthonormality of $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, we also have

$$\mathsf{a}^{(1)} + \mathsf{a}^{(2)} + \mathsf{a}^{(3)} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}.$$

Using (30) in the case of Fig. 10, the material frame orientation tensor for the tows is given by

$$\mathsf{a}^{(1)} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathsf{a}^{(2)} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathsf{a}^{(3)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{31}$$

The tows are thus isotropically oriented in the X-Y plane for $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, and unidirectionally oriented in the Z axis for $\boldsymbol{e}_3$. For the matrix phase in Fig. 10, its material frame coincides with the global frame. In this case, a is similar to unidirectional orientation tensors and satisfies

$$\mathsf{a}^{(i)}_{ii} = 1, \quad \mathsf{a}^{(i)}_{jk} = 0 \quad \text{for other components.} \tag{32}$$

Suggested by [27], such orientation tensors can also be defined for DMN. As explained in Sect. 2.1, DMN rotations $\boldsymbol{\theta}$ define rotation matrices between the local frame of the current laminate to that of the next nesting level. They can thus be composed to obtain the effective rotation matrix from the material frame $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ (leaf laminates) to the global frame of the microstructure Ω (root laminate). Let $\mathsf{p}(i)$ denote the parent of a laminate i in the DMN binary tree architecture. For instance, in the 3-layer DMN example shown in Fig. 2, we have $\mathsf{p}(4) = 2$ for the leaf laminate 4 and $\mathsf{p}^2(4) = \mathsf{p}\big(\mathsf{p}(4)\big) = \mathsf{p}(2) = 1$ which is the root laminate. For each leaf laminate i which carries the DMN material nodes, the effective rotation matrix from the material frame to the global one is given by

$$\tilde{\boldsymbol{R}}_i = \boldsymbol{R}(\boldsymbol{\theta}_{\mathsf{p}^{n-1}(i)}) \cdots \boldsymbol{R}(\boldsymbol{\theta}_{\mathsf{p}^2(i)}) \boldsymbol{R}(\boldsymbol{\theta}_{\mathsf{p}(i)}) \boldsymbol{R}(\boldsymbol{\theta}_i), \quad \tilde{\boldsymbol{R}}_i \in \mathrm{SO}(3). \tag{33}$$

Note that for an n-layer DMN, we have necessarily $\mathsf{p}^{n-1}(i) = 1$ for arbitrary leaf laminate i. Due to the absence of input rotation matrices for $(\mathbb{C}_1, \mathbb{C}_2)$, material nodes $1 \leq i \leq \mathsf{n}$ that share the same leaf laminate also obtain the same effective rotation matrix. For instance, for the material nodes 3 and 4 contained in the leaf laminate 5 in Fig. 2, their effective rotation is

$$\tilde{\boldsymbol{R}}_5 = \boldsymbol{R}(\boldsymbol{\theta}_1)\boldsymbol{R}(\boldsymbol{\theta}_2)\boldsymbol{R}(\boldsymbol{\theta}_5).$$

Using these effective rotations on leaf laminates, similar to (30), the DMN material frame orientation tensor can be computed for the phase 1

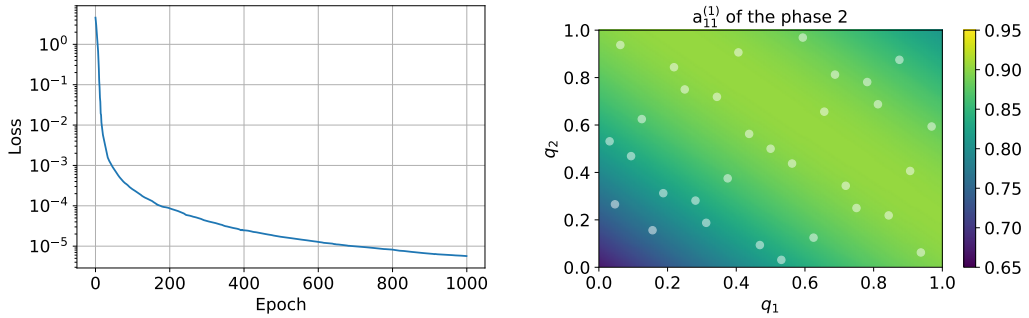
$$\mathsf{a}^{(i)}_{\mathrm{DMN}}(\boldsymbol{w}, \boldsymbol{\theta}) = \frac{\sum_{i \in \mathbb{I}_1} w_i \tilde{\boldsymbol{e}}_i \otimes \tilde{\boldsymbol{e}}_i}{\sum_{i \in \mathbb{I}_1} w_i}. \tag{34}$$

Similar formula can be defined for the phase 2. Compared to the DMN volume fraction (26), the computation of DMN orientation tensors requires both DMN weights $\boldsymbol{w}$ and rotations $\boldsymbol{\theta}$. Using the PINN for DMN parameters (25), we propose the following orientation constraint through the definition of a loss function

$$\mathcal{L}_{\mathsf{a}} = \frac{1}{n}\sum_{i=1}^{n} \|\mathsf{a}_{\mathrm{DMN}}\big(\boldsymbol{w}(\boldsymbol{p}_i), \boldsymbol{\theta}(\boldsymbol{p}_i)\big) - \mathsf{a}(\boldsymbol{p}_i)\|^2, \quad \|\mathsf{a}\|^2 = \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\left(a^{(i)}_{jk}\right)^2. \tag{35}$$

Similar loss function is defined for the phase 2 and then summed together. In (35), $\boldsymbol{p}_i$ are the collocation points in the microstructural parametric space and $\mathsf{a}(\boldsymbol{p}_i)$ are the orientation tensors of the parametric microstructure. The training strategy of (35) will be described further in Sect. 3.5.

With 5 DMN layers and the ReLU activation function, an example of the minimization of (35) is shown in Fig. 11. The microstructure is parametrized by two geometrical parameters $\boldsymbol{p} = \boldsymbol{q} = (q_1, q_2) \in [0,1]^2$ that do not change vf. Due to (25a), the DMN weights do not vary with $\boldsymbol{p}$. We suppose that the material frame of both phases coincides with the global one, hence in (35) the unidirectional orientation tensors (32) are used as target values. The decreasing loss function value indicates that our PINN (25) is capable of satisfying the orientation constraint.



(a) (b)

Figure 11: Orientation constraint prescribed on the PINN for $\boldsymbol{w}(\boldsymbol{p})$ and $\boldsymbol{\theta}(\boldsymbol{p})$: (a) minimization of the loss function (35) and (b) material frame orientation tensor component $\mathsf{a}^{(1)}_{11}$ for the 1st material axis $\boldsymbol{e}_1$ of the phase 2, at epoch 10 of training. The points in (b) represent the $\mathsf{n} = 2^5 = 32$ collocation points.

At the early stage of the training, $\mathsf{a}^{(1)}_{11}$ of the phase 2 still presents variations in the parametric space (q_1, q_2). Its mean and standard deviation values at epoch 10^1 and at epoch 10^3 are indicated in Tab. 3. After training, $\mathsf{a}^{(1)}_{11} \approx 1$ becomes quasi-uniform as expected.

$\mathsf{a}^{(1)}_{11}$	Mean	Standard deviation
At epoch 10^1	0.866	0.0440
At epoch 10^3	0.999	0.000686

Table 3: Mean and standard deviation of $\mathsf{a}^{(1)}_{11}$ of the phase 2 in the parametric space at epoch 10^1 and at epoch 10^3.

3.5. Offline training of PINN-DMN

As shown in Fig. 7, the PINN-DMN fitting parameters in (25) are optimized *jointly* using the linear elastic behavior data at different microstructural parameters. This training strategy is also used by [11] for short-fiber reinforced plastics with fixed volume fraction but arbitrary fiber orientations.

Each sample s in the training data contains the homogenized linear elastic stiffness tensor $\overline{\mathbb{C}}_s^{\text{FE}}$ of a particular microstructure with parameters $\boldsymbol{p}_s$, given input linear elastic behaviors of both phases $(\mathbb{C}_1, \mathbb{C}_2)_s$, see Fig. 1. Recall that $\overline{\mathbb{C}}_s^{\text{FE}}$ is in generally performed by computational homogenization. The generation of such synthetic dataset requires sampling both in the input material space and in the parametric space of the microstructure

$$(\mathbb{C}_1, \mathbb{C}_2)_s \in \mathbb{M}, \quad \boldsymbol{p}_s = (\text{vf}, \boldsymbol{q})_s \in \mathbb{P} \subset \mathbb{R} \times \mathbb{R}^q \quad \Longrightarrow \overline{\mathbb{C}}_s^{\text{FE}}. \tag{36}$$

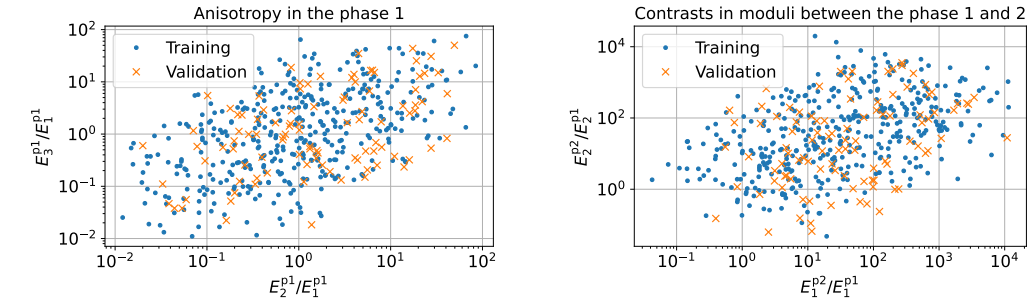
In (36), $\mathbb{M}$ represents the discrete material sample set containing various input material properties of $(\mathbb{C}_1, \mathbb{C}_2)$, while $\mathbb{P}$ is the parametric sample set. In this work, these two samplings are performed *independently*. For different microstructural parameter values, the same material sampling is used to evaluate the linear elastic homogenization behavior of the microstructure

Material sampling. We follow the original material sampling method proposed by [9], assuming that $(\mathbb{C}_1, \mathbb{C}_2)$ are both orthotropic in their respective material frames. In total, $9 + 9 = 18$ material parameters are required to characterize their orthotropic elastic behaviors and 1 additional scaling parameter is used to introduce contrasts in the elastic moduli between the two phases. In this work, Latin hypercube sampling [42] is used to sample this 19-dimensional space. The stiffness tensors of the phase 1 are also normalized since the homogenization function (1) that DMN attempts to learn is homogeneous

$$(\mathbb{C}_1, \mathbb{C}_2) \mapsto \overline{\mathbb{C}} \implies (k\mathbb{C}_1, k\mathbb{C}_2) \mapsto k\overline{\mathbb{C}} \quad \forall k \in \mathbb{R}.$$

This property is satisfied by the DMN homogenization function (4) thanks to its micromechanics-based architecture.

After sampling, the generated $(\mathbb{C}_1, \mathbb{C}_2)_i$ are randomly partitioned into a training set and a validation set. An example of material sampling is given in Fig. 12, for the unidirectional fiber composite example in Sect. 4.1. Anisotropy in the phase 2 is similar to Fig. 12(a) and is not shown. In this case, since the fibers are much stiffer than the matrix, the scaling parameter is adapted to generate appropriate contrasts in the elastic moduli between the two phases. From Fig. 12(b), it can be seen that ratios between the Young's moduli range from 10^{-1} to 10^4.



(a) (b)

Figure 12: Material sampling for input orthotropic stiffness tensors $(\mathbb{C}_1, \mathbb{C}_2)$. Both phases present anisotropy in their elastic behaviors and contrasts in the elastic moduli are also introduced between them.

The distance from isotropy of a given stiffness tensor can characterize its degree of anisotropy, see [43]. Using the L_2 (Frobenius) norm of elasticity tensors, the following relative distance from isotropy can be defined

$$d_{\text{iso}} = \frac{\|\mathbb{C} - \mathbb{C}_{\text{iso}}\|}{\|\mathbb{C}\|}, \tag{37}$$

where $\mathbb{C}_{\text{iso}}$ is the closest isotropic elasticity tensor of $\mathbb{C}$ in the Frobenius norm. An explicit formula is provided in [43]. If $d_{\text{iso}} = 0$, it implies that $\mathbb{C}$ is isotropic. In Fig. 13, the statistical distributions of d_{iso} are computed on the total dataset. Similar histogram is obtained for both phases. Anisotropy is not uniformly distributed in the sampling space. The median value of d_{iso} is approximately 0.6, while the maximum value is around 0.8.

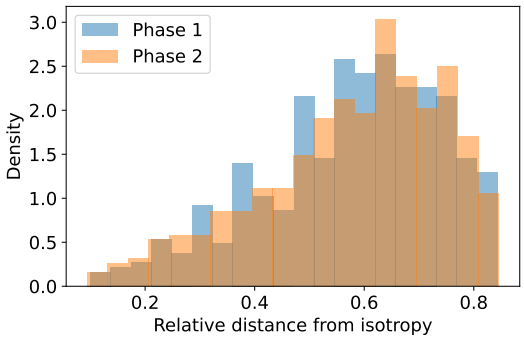


Figure 13: Anisotropy in both phases characterized by their relative distance from isotropy.

Sampling in the microstructural parametric space. Two different sampling needs to be performed in the microstructural parametric space. The first one defines the parameters values $\boldsymbol{p}_s \in \mathbb{P}$ in (36) on which $\overline{\mathbb{C}}$ is computed together with the previous material sampling $\mathbb{M}$. For different microstructures, different sampling strategy may be used to well cover the region of interest in the parametric space. For instance, in [11], the fiber orientation space is sampled by subdividing recursively the orientation triangle into sub-triangles. For microstructures depending solely on the volume fraction parameter [26], equidistant sampling can be used in the vf-interval of interest. In general, all the microstructural parameters $\boldsymbol{p} = (\mathrm{vf}, \boldsymbol{q})$ are sampled jointly in order that less sampling points are needed to optimally fill the parametric space.

The other sampling concerns the physical constraints prescribed on PINN (25). The volume fraction constraint (28) and the orientation constraint (35) are weakly enforced using collocation points in the parametric space. Since the costly computational homogenization is not carried out for these points, more collocation points can be used.

The volume fraction constraint (28) requires a sampling of the vf parameter. A uniform sampling in $[0, 1]$ is simply used to generate them: $\mathrm{vf}_i = i/(n-1)$, with $i = 0, 1, \ldots, n-1$. The number of collocation points is taken to be the number of DMN material nodes n. A numerical study not reported here shows that n is sufficient to prescribe (27) with an acceptable accuracy, which is demonstrated by the example presented in Sect. 3.3.

The orientation constraint (35) requires collocation points $\boldsymbol{p}_i$ in the parametric space, since it is prescribed both on DMN weights and rotations. Assume that each microstructural parameter is rescaled such that $\boldsymbol{p} \in [0, 1]^{q+1}$. Sobol low-discrepancy sequence [44] is used to generate these collocation points which presents good space filling properties. If internal bounds within the parameters are present, such as for the fiber orientation space [11], collocation points outside the feasible region are removed. The number of collocation points is also taken to be the number of DMN material nodes n. Sobol sequence is deterministic so that the same collocation points are used given the number of layers n. Numerical example shown in Sect. 3.5 shows that the orientation constraint can be well prescribed with this sampling strategy.

Loss function. Based on the previous material sampling and samplings in the parametric space, a total loss function $\mathcal{L}$ as defined as follows

$$\mathcal{L} = \frac{1}{|\mathbb{P}|} \sum_{\mathbb{P}} \mathcal{L}_{\boldsymbol{p}} + \mathcal{L}_{\mathrm{vf}} + \mathcal{L}_{\mathsf{a}}, \qquad \mathcal{L}_{\boldsymbol{p}} = \frac{1}{|\mathbb{M}|} \sum_{\mathbb{M}} e_i^2, \qquad e_i = \frac{\|\overline{\mathbb{C}}_i^{\mathrm{DMN}} - \overline{\mathbb{C}}_i^{\mathrm{FE}}\|}{\|\overline{\mathbb{C}}_i^{\mathrm{FE}}\|} \tag{38}$$

In (38), $\mathcal{L}_{\mathrm{vf}}$ is the volume fraction constraint (28) and $\mathcal{L}_{\mathsf{a}}$ is the orientation constraint (35). Scaling factors are absent in front of these two terms since our numerical simulations indicate that they are not required. In [9, 10], a penalty term is introduced to control the magnitude of $\boldsymbol{z}$ in (5). It is motivated by the fact that the ReLU activation function is unbounded and the DMN homogenization function (4) is invariant with respect to the scaling $\boldsymbol{z} \mapsto k\boldsymbol{z}$ for $k > 0$. In this work, we don't find it necessary to prescribe such constraints on our fitting parameters $(\boldsymbol{w}_0, \boldsymbol{w}_1)$.

Similar to [9], the loss at a fixed microstructural parameter value $\mathcal{L}_{\boldsymbol{p}}$ is the mean squared error in the material sample set $\mathbb{M}$ comparing the DMN predictions and the computational homogenization ones, with the relative L_2 norm on the stiffness tensors. In [10, 11], the L_1 norm is used to compute each e_i, which is then aggregated into $\mathcal{L}_{\boldsymbol{p}}$ with the L_{10} norm. This is not tested in this work. In the total loss, $\mathcal{L}_{\boldsymbol{p}}$ is then averaged among different samples of the microstructural parameters.

A separate numerical study not reported here investigated the log-Euclidean distance [43] in computing e_i. The log-Euclidean distance provides an intrinsic distance between two symmetric positive definite matrices, since the same distance is obtained by using the stiffness tensor or the compliance tensor. However, our numerical results

show that the prediction in the homogenized Poisson ratios may become less accurate. We suspect that it is due to the fact that the distance is now computed on matrix logarithms.

Optimization algorithm. The total loss function (38) is minimized using gradient-based methods, similar to other DMN formulations [7, 10, 18, 19]. Our PINN-DMN architecture is implemented using PyTorch [45]. The derivatives of the loss function (38) with respect to the fitting parameters (25) can thus be easily computed with automatic differentiation. Single precision is used for offline training.

Our PINN-DMN forward function $(\boldsymbol{p}, \mathbb{C}_1, \mathbb{C}_2) \mapsto \overline{\mathbb{C}}$ is vectorized over both the material sampling dimensions $(\mathbb{C}_1, \mathbb{C}_2) \in \mathbb{M}$ and the parameter sampling dimension $\boldsymbol{p} \in \mathbb{S}$, in order to achieve optimal parallel computational efficiency. In [7, 11, 18, 19], a mini-batch (or even each sample) is randomly drawn from the whole dataset at each iteration of an epoch to introduce randomness in gradient descent and promote more frequent parameter updates. In the machine learning community, this is known to improve generalization capability [46]. In this work however, the entire dataset is trained in a single batch at every epoch. Our simulation results indicate that the trained PINN-DMN still generalizes well without significant overfitting, thanks to its physics-based nature. Furthermore, the training time can be reduced due to vectorization.

We have compared different optimizers provided in PyTorch, and have found that the resilient backpropagation (Rprop) algorithm [47] works best in terms of convergence (loss values), stability (noise) and efficiency (time to train each epoch) in this single batch setting. In general, our PINN-DMN is trained over 10000 epochs with an initial learning rate of 10^{-2}.

Initialization. Depending on the activation function used in (25a), different initialization can be used for $\boldsymbol{w}_0$. For ReLU, uniform distribution $\boldsymbol{w}_0 \sim \mathcal{U}(0.2, 0.8)$ as proposed by [7] is applied. For $\boldsymbol{w}_1$ which characterizes the dependence of the DMN weights on vf, it is zero-initialized. Compared to a random initialization, numerical simulations demonstrate that it would result in better convergence of the loss function.

Regarding the DMN rotations in (25b), similarly $\boldsymbol{\Theta}_1$ is zero-initialized. The dependence of $\boldsymbol{p} \mapsto \boldsymbol{\theta}(\boldsymbol{p})$ is hence also learned from zero. For the constant part $\boldsymbol{\theta}_0$, a random initialization on quaternions is used. Each component is initialized using the standard normal distribution, and then normalized to obtain a unit norm. We don't find it necessary to control the norm of each DMN rotation $\boldsymbol{\theta}$ in (25b) through the use of an additional loss term in (38).

Since initialization can have an impact on trained DMN parameters, training is in general repeated 20 times and the one with the least final loss value is chosen for further numerical investigations.

4. Numerical examples

In this section we will numerically evaluate our PINN-DMN architecture on three parametric microstructures. Due to the amount of training data requiring in total $|\mathbb{P}| \times |\mathbb{M}|$ FE-RVE simulations, training is performed on an NVIDIA GeForce RTX 2080 Ti GPU card using single precision.

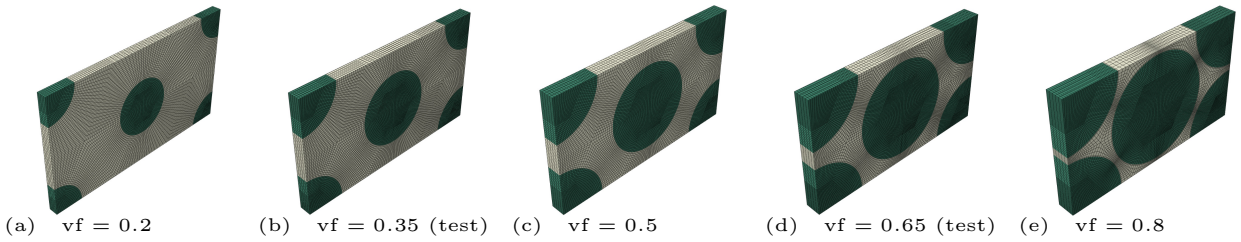
4.1. Unidirectional fiber composite

We first consider a unidirectional fiber composite with varying fiber volume fractions. The fully-connected architecture is compared with the physics-informed one in terms of accuracy and generalization ability. The influence of the physical constraints (28) and (35) is also discussed. Since there is only one microstructural parameter, the transfer-learning based approach is also tested and then compared with PINN-DMN.

The 3-d finite element model is built with hexagonal fiber packing using the FE-RVE plugin available in Abaqus CAE [48]. In total, 5 FE-RVE models are constructed, see Fig. 14. Three of them (vf = 0.2, vf = 0.5 and vf = 0.8) constitute the parametric sample set $\mathbb{P}$ and are used to generate training dataset, while the other two are used to test interpolation accuracy. The FE-RVE analysis is conducted by imposing periodic displacement on the boundary nodes. The 6×6 linear elastic stiffness tensor is obtained by performing a linear perturbation step in Abaqus in which 6 different load cases are applied.

Material sampling is performed using the method described in Sect. 3.5. In total, the same 500 input orthotropic material properties are sampled for all the five microstructures, see Fig. 12. Since the actual composite is made of a polypropylene matrix reinforced with glass fibers, the synthetic fiber stiffnesses $\mathbb{C}_2$ are also generated to be statistically higher than the matrix ones $\mathbb{C}_1$. The real linear elastic properties of the two isotropic phases are indicated in Tab. 4.

For training microstructures vf = 0.2, vf = 0.5 and vf = 0.8, 400 of the 500 samples are used as training dataset, while the other 100 are reserved for validation. For the others vf = 0.35 and vf = 0.65, all the 500 samples are used for testing the interpolation accuracy. The simulation is run with 24 threads using Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz. The FE-RVE dataset generation time is summarized in Tab. 5.



(a) vf = 0.2 (b) vf = 0.35 (test) (c) vf = 0.5 (d) vf = 0.65 (test) (e) vf = 0.8

Figure 14: FE-RVE models for the unidirectional fiber composite with varying fiber volume fractions. The models with vf = 0.2, vf = 0.5 and vf = 0.8 are used to generate training dataset, while those with vf = 0.35 and vf = 0.65 are used to test interpolation accuracy.

	E (MPa)	ν
Matrix	3300	0.41
Fiber	72000	0.22

Table 4: Real linear elastic properties of the two phases for the unidirectional fiber composite.

Since the volume fraction (of the fibers) is the only parameter of this microstructure, the PINN-DMN architecture (25) implies that the DMN weights $\boldsymbol{w}$ vary with vf according to (25a), while the DMN rotations $\boldsymbol{\theta}$ remain *constant*. The fitting parameters are hence $(\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{\theta}_0)$. This is somehow similar to the "naïve" approach described in [26]. Given a pre-trained base DMN with $\boldsymbol{w}^{(\mathrm{b})}$ and corresponding volume fraction $\mathrm{vf}^{(\mathrm{b})}$, the DMN weights at a new vf are scaled appropriately based on the volume fraction

$$\boldsymbol{w}_i = \begin{cases} \dfrac{1-\mathrm{vf}}{1-\mathrm{vf}^{(\mathrm{b})}} \boldsymbol{w}_i^{(\mathrm{b})} & i \in \mathbb{I}_1, \\ \dfrac{\mathrm{vf}}{\mathrm{vf}^{(\mathrm{b})}} \boldsymbol{w}_i^{(\mathrm{b})} & i \in \mathbb{I}_2. \end{cases} \tag{39}$$

The DMN rotations also remain unchanged. Compared to (39), the vf-dependence of $\boldsymbol{w}$ in our PINN-DMN is not directly prescribed but learned from data.

Training of our PINN-DMN is conducted following Sect. 3.5, with the two physical constraints (28) and (35). The ReLU activation function is used. Since the material frames of both phases coincide with the global one, the unidirectional material frame orientation tensor (32) is used as the target value for the orientation constraint. With 5 DMN layers, we first compare the Rprop optimizer with the frequently used Adam optimizer [49] in terms of loss history, in Fig. 15. A total of 20 trainings are realized using random initialization for each optimizer. With Adam, the loss history is much more noisy (even using a single batch) and converges slower than Rprop. The loss at the end of training (10000 epochs) is also slightly lower statistically with the Rprop optimizer. Rprop is hence chosen as the optimizer of PINN-DMN.

Similar to the original DMN formulations [7], adding more layers also increases the expressive power of PINN-DMN. From Fig. 16(a), the final loss values with 7 layers are lower than those obtained with 5 layers. The total loss can be partitioned into two parts according to (38): the FE-RVE data part and the physical constraints part. Their respective history is shown in Fig. 16(b) for the training realization with the least final loss value, for 7 layers. Each of the three "FE-RVE" curves represents one particular training microstructure vf = 0.2, vf = 0.5 or vf = 0.8, and is monotonically decreasing simultaneously with similar values. The physical constraints part also converges well and is lower than the FE-RVE part by an order of magnitude. Among the 20 training realizations, the median

FE-RVE model	Number of degrees of freedom	Dataset generation time
vf = 0.2	131568	$500 \times 3 = 1500$ s
vf = 0.35	169368	$500 \times 4 = 2000$ s
vf = 0.5	164328	$500 \times 4 = 2000$ s
vf = 0.65	208752	$500 \times 5 = 2500$ s
vf = 0.8	247488	$500 \times 6 = 3000$ s

Table 5: FE-RVE dataset generation time for each unidirectional fiber composite at different fiber volume fractions.

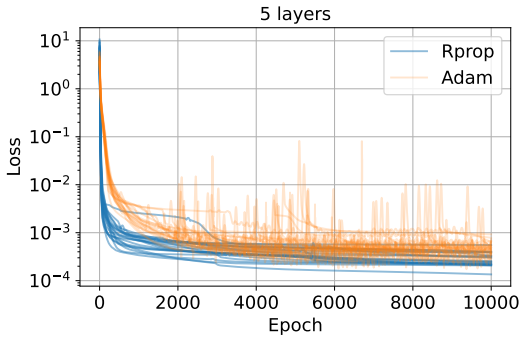
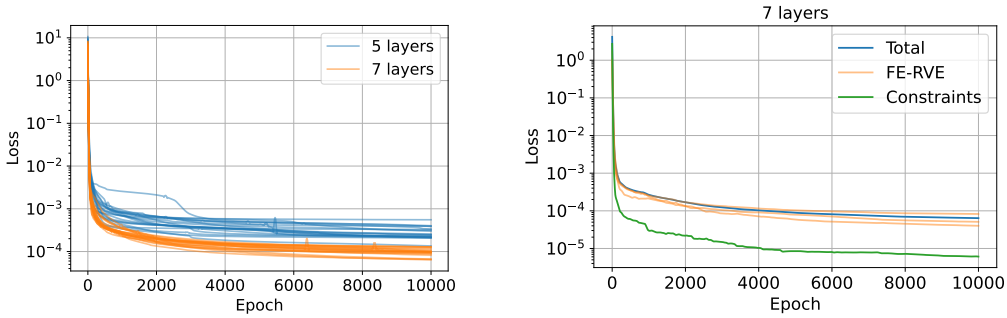


Figure 15: Loss history with 5 DMN layers with the Rprop and the Adam optimizers.

training time for 5 or 7 layers is indicated in Tab. 6. In the sequel, we will report the results obtained with 5 layers which provides satisfactory accuracy.



(a) (b)

Figure 16: PINN-DMN training for the unidirectional fiber composite: (a) loss histories with 5 and 7 DMN layers; (b) partition of the total loss into the FE-RVE data part and the physical constraints part.

	5 layers	7 layers
Median training time	570 s	786 s

Table 6: Median training time with 5 or 7 DMN layers for the unidirectional fiber composite.

In Fig. 17, the statistical distribution of the relative error e_i in (38) between DMN predictions and FE-RVE ones at each material sample is shown for the microstructure with vf $= 0.5$. The 0.1, 0.5 (median) and 0.9-quantiles are also indicated. Even though the maximum relative error can reach 6% for some localized samples, the error is less than 2.4% for 90% of them. We believe that these quantiles are less sensitive to outliers and hence more appropriate for comparing different models.

Influence of physical constraints and parametric DMN architectures

We first analyze the influence of physical constraints on the PINN-DMN architecture (25). The training, validation and test errors are compared in Fig. 18 using their respective 0.1, 0.5 (median) and 0.9-quantiles at different volume fractions. The PINN-DMN model without physical constraints is trained similarly with 20 realizations and the one with the least final loss value is chosen. Validation error refers to the relative error computed on the validation dataset composed of 100 material samples. Overfitting is not observed for both cases, since the validation errors are comparable to the training errors. When the physical constraints are not included, the training errors at vf $= 0.5$ and vf $= 0.8$ are similar or only slightly lower at vf $= 0.2$ than the case when they are considered. However, the inclusion of the physical constraints reduces the interpolative test errors at vf $= 0.35$ and vf $= 0.65$ for previously unseen microstructures. The physical constraints may hence improve generalization ability of PINN-DMN.

In Fig. 19, we analyze the influence of the physical constraints on the fully-connected architecture (23). Now, the DMN rotations also become a function of the volume fraction, leading to an additional fitting parameter $\boldsymbol{\Theta}_1$ during

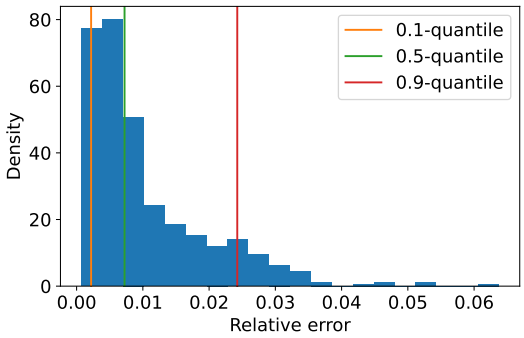
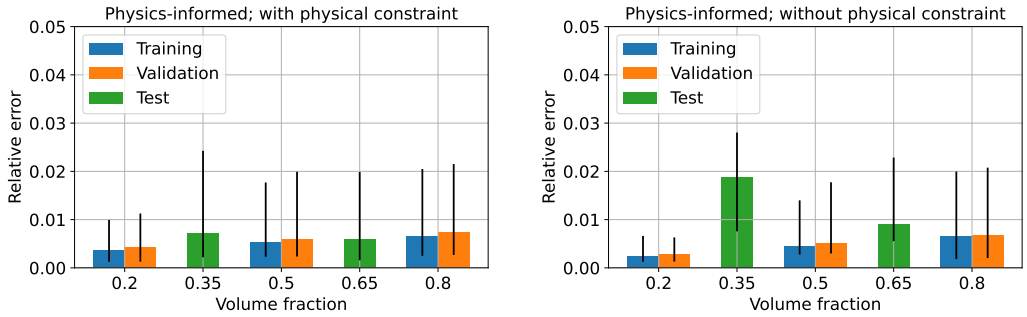


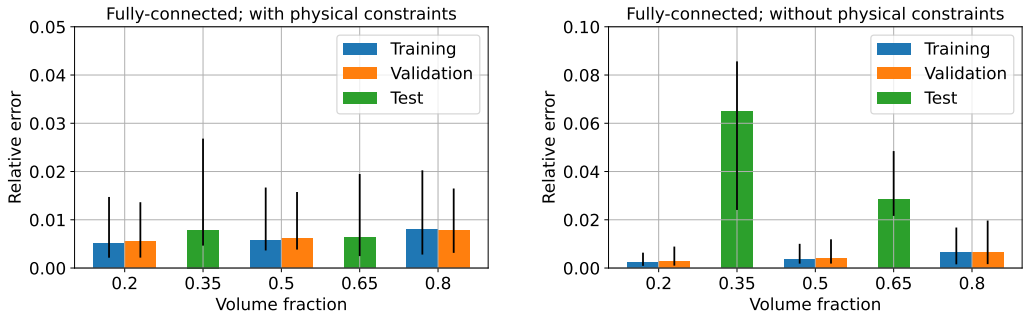
Figure 17: Statistical distribution of the relative error between DMN predictions and FE-RVE ones at each material sample, for the microstructure with vf = 0.5.



(a) (b)

Figure 18: Training, validation and test errors of PINN-DMN at different volume fractions represented by their respective 0.1, 0.5 (median) and 0.9-quantiles: (a) with physical constraints; (b) without physical constraints.

training. The physical constraints reduce overfitting at the data points. By comparing Fig. 18(a) and Fig. 19(a), we observe similar training, validation and test errors between the PINN and the fully-connected architectures, when the physical constraints are included. The PINN architecture (25) hence demonstrates similar expressive power compared to the fully-connected one, even with a constant DMN rotations vector $\boldsymbol{\theta}_0$ for different volume fractions. When the physical constraints are not considered, the interpolative test errors become significantly larger (even the scale of the y-axis needs to be changed) in Fig. 19(b) for the fully-connected architecture. Comparing Fig. 18(b) and Fig. 19(b), we conclude that the PINN architecture itself (by requiring a constant DMN rotation vector $\boldsymbol{\theta}_0$) may help to reduce overfitting in the parametric space and improve generalization capability.

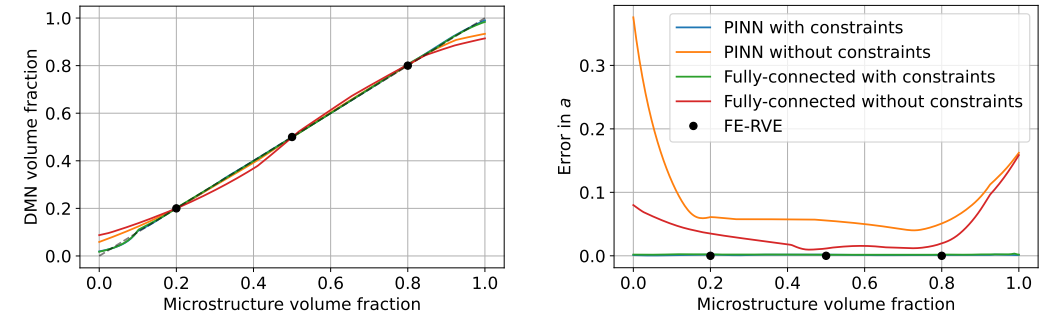


(a) (b)

Figure 19: Training, validation and test errors of a fully-connected parametric DMN at different volume fractions represented by their respective 0.1, 0.5 (median) and 0.9-quantiles: (a) with physical constraints; (b) without physical constraints.

In Fig. 20(a), the DMN volume fraction prediction (26) is presented as a function of the microstructure volume fraction. For all cases, DMN recovers well the volume fractions 0.2, 0.5 and 0.8 at the FE-RVE data points. The microstructure morphologies are hence well learned by DMN using linear elastic data [7, 10]. However, away from

these three data points, the DMN volume fraction prediction may differ from the actual microstructure one. At $\mathrm{vf} = 0.35$, when (28) is explicitly included during training, the relative errors $|\mathrm{vf}_{\mathrm{DMN}} - \mathrm{vf}_{\Omega}|/\mathrm{vf}_{\Omega}$ are smaller than 0.4% for the two parametric DMN architectures. However, when it is not the case, we obtain 2% with PINN and even 9% with the fully-connected architecture. This may explain the higher interpolative testing errors at $\mathrm{vf} = 0.35$ for these two models in Fig. 18(b) and Fig. 19(b). The straight line (27) can only be well recovered when the volume fraction constraint (28) is explicitly included.



(a) (b)

Figure 20: Physical constraints verification for two parametric DMN architectures: (a) DMN volume fraction prediction; (b) error on the material frame orientation tensor.

Similarly, the error on the material frame orientation tensor $\|\mathsf{a}_{\mathrm{DMN}}(\mathrm{vf}_i) - \mathsf{a}(\mathrm{vf}_i)\|$ using the norm defined in (35) is presented in Fig. 20(b). The errors on the fibers and on the matrix are summed together. When (35) is not included during training, deviation from the theoretical unidirectional material frame orientation tensor (32) can be observed, even though the error is well bounded between $[0.2, 0.8]$. Similar to Fig. 20(a), physical constraint errors begin to increase significantly when extrapolating outside the training domain. The inclusion of (35) may hence improve the generalization capability in the whole parametric space.

In this work, the rotation matrix for $(\mathbb{C}_1, \mathbb{C}_2)$ for each of the $\mathsf{n} = 2^n$ material nodes on the input layer is omitted compared to the original formulation [7, 9]. In Fig. 21, the training, validation and test errors are shown when they are instead included. Compared to Fig. 18(a), the inclusion of such material rotation matrices does not increase the expressive power of PINN-DMN. This could be partially due to the fact that the microstructure being considered does not contain local material orientation.

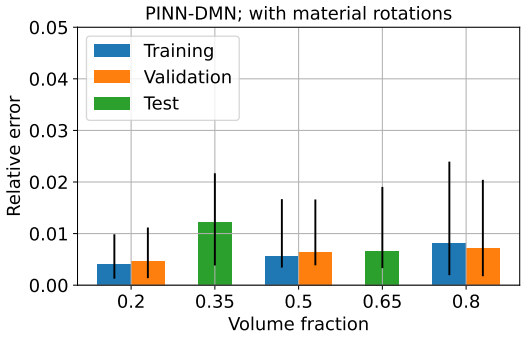


Figure 21: Training, validation and test errors of PINN-DMN with additional $\mathsf{n} = 2^n = 32$ material rotations at different volume fractions represented by their respective 0.1, 0.5 (median) and 0.9-quantiles.

Using the real linear elastic properties of the two phases in Tab. 4, the relative errors between the DMN predictions and the FE-RVE result for two parametric DMN architectures and with or without the physical constraints are shown in Fig. 22. The conclusions drawn from Fig. 18 and Fig. 19 are recovered. The PINN architecture and the physical constraints both help to improve generalization ability of the parametric DMN model. Using PINN-DMN with the physical constraints, the relative errors are less than 2% for the microstructures being considered.

The homogenized elastic property prediction is presented in Fig. 23 in the *whole* parametric space. Since the effective elastic tensor is orthotropic, it can be reduced to 9 elastic moduli. The global 1 direction refers to the fiber longitudinal direction. The general nonlinear influence of the volume fraction is well captured by these two

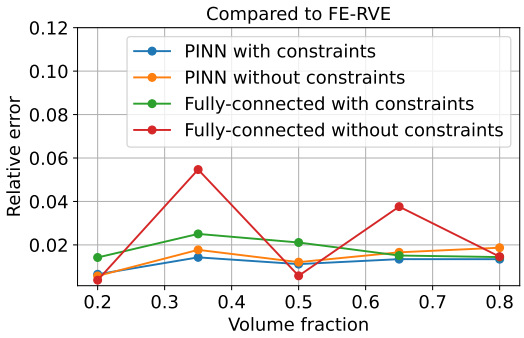


Figure 22: Relative error using real properties between the DMN predictions and the FE-RVE results for two parametric DMN architectures and with or without the physical constraints.

parametric DMN models. The use of a PINN architecture as well as the physical constraints helps to improve both *interpolative* and *extrapolative* generalization capabilities.

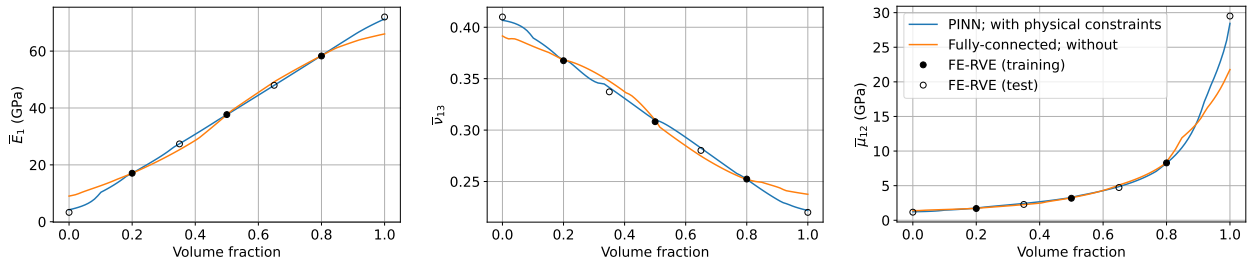


Figure 23: Homogenized elastic moduli prediction as a function of the fiber volume fraction obtained by parametric DMN models: (a) Young's modulus $\overline{E}_1$, (b) Poisson ratio $\overline{\nu}_{13}$ and (c) shear modulus $\overline{\mu}_{12}$. The FE-RVE training and test data are indicated. The theoretical values at vf $= 0$ and vf $= 1$ are also provided as test data.

In Fig. 24(a), the PINN functional dependence of the DMN weights (25a) is illustrated. The PINN-DMN architecture with the physical constraints is used. Globally, 22 of the 32 material nodes are active. With the increasing fiber volume fraction, DMN weights of the matrix decrease *gradually* and *individually*, while those of the fiber increase at the same time. This is also reflected in Fig. 24(b), where the ratio of the active DMN nodes is presented for both phases. Even though the ratios for both phases gradually increase or decrease, the total number of active nodes does not vary much. This ensures the expressive power of PINN-DMN in the whole parametric space. The functional dependence on vf is realized both by adapting weights values and the number of active nodes for each phase.

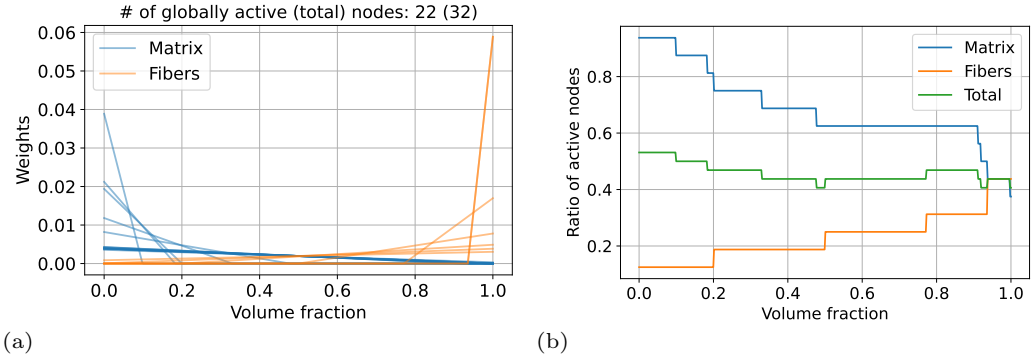
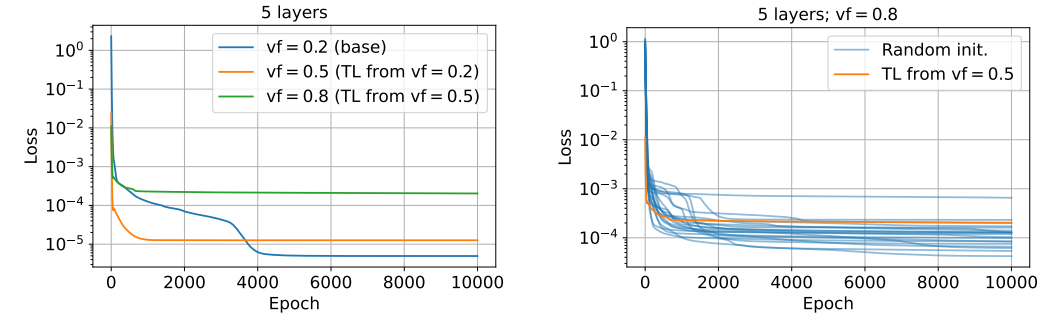


Figure 24: PINN-DMN: (a) DMN weights evolution with the volume fraction; (b) variation of the ratio of active DMN nodes

Comparison with transfer-learning based interpolative DMN

We follow the procedure described in [26] and the microstructure with the least fiber volume fraction is used as the starting point in the training sequence $\mathrm{vf}_0 = 0.2 \rightarrow \mathrm{vf}_1 = 0.5 \rightarrow \mathrm{vf}_2 = 0.8$. Subsequent training is initialized using (39) based on the previously trained DMN. As for our parametric DMN models, the transfer learning process is repeated 20 times with random initialization when training the base DMN model. The realization with the least final total loss value $\mathcal{L}_{\mathrm{tot}} = \mathcal{L}_0 + \mathcal{L}_1 + \mathcal{L}_2$ is then chosen for further analysis. For comparison with previous PINN-DMN results, 5 DMN layers are used.

In Fig. 25(a), loss histories obtained with transfer learning are presented. Transfer learning indeed accelerates convergence. For $\mathrm{vf} = 0.5$ and $\mathrm{vf} = 0.8$, convergence is achieved within the first 1000 epochs. However, the final loss values become an order of magnitude larger at $\mathrm{vf} = 0.8$ compared to the base DMN. In [26], there is also a slight increase in the final loss values at the end of the transfer learning process. For $\mathrm{vf} = 0.8$, DMN is also trained using random initialization. The loss histories are compared with that obtained from transfer learning. A reduction of the expressive power of DMN is hence observed with transfer learning.



(a) (b)

Figure 25: (a) Loss histories for different volume fractions using transfer learning; (b) loss history at $\mathrm{vf}_2 = 0.8$ obtained using random initialization and from transfer learning.

The training, validation and test errors obtained with this transfer-learning based interpolative DMN model are presented in Fig. 26. Compared to the results obtained with PINN-DMN (with the physical constraints) in Fig. 18(a), the errors at the training points $\mathrm{vf} = 0.2$ and $\mathrm{vf} = 0.5$ are lower with this transfer learning approach. However, due to the reduction of the expressive power, the errors at $\mathrm{vf} = 0.8$ become comparable to those with PINN-DMN. The test errors are significantly larger than the training / validation ones, especially at $\mathrm{vf} = 0.65$. The transfer-learning may hence tend to overfit at the data points.

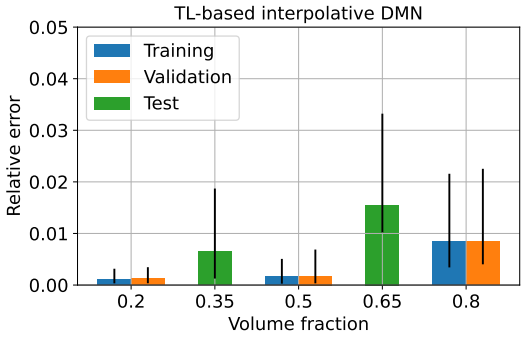


Figure 26: Transfer-learning based interpolative DMN: training, validation and test errors at different volume fractions represented by their respective 0.1, 0.5 (median) and 0.9-quantiles.

Even though it is not proposed in [26], we can also use (39) to extrapolate DMN weights outside the training domain. In this case, the DMN weights at $\mathrm{vf} = 0$ are hence extrapolated from the base DMN at $\mathrm{vf} = 0.2$, while the DMN trained by transfer-learning at $\mathrm{vf} = 0.8$ can be used to obtain the DMN weights at $\mathrm{vf} = 1$. Since (39) respects the actual volume fraction (26), the volume fraction constraint $\mathrm{vf}_{\mathrm{DMN}} = \mathrm{vf}_\Omega$ is satisfied at all vf when combined with piecewise linear interpolation. However, it is not the case for the orientation constraint. In Fig. 27, the error on the material frame orientation tensor is computed in the whole parametric space. Compared to PINN-DMN, the relative error is much larger with the transfer-learning approach. The error also increases with interpolation.

At the test points vf = 0.35 and vf = 0.65, the errors become approximately 2% and 3%. This may explain the larger test errors in the transfer-learning model in Fig. 26.

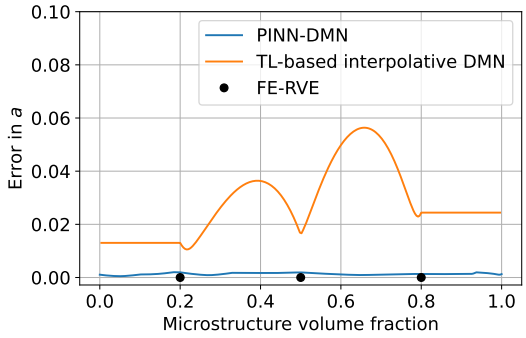


Figure 27: Error on the material frame orientation tensor obtained by PINN-DMN with the physical constraints and by the transfer-learning approach.

The relative errors using the real linear elastic properties of the two phases in Tab. 4 are shown in Fig. 28 for the PINN-DMN model and the transfer-learning (TL) approach. Even though the TL approach is more accurate for the first 3 volume fractions, it becomes less accurate for the last 2 volume fractions. The relative errors obtained with PINN-DMN, on the contrary, are more uniform in the parametric space.

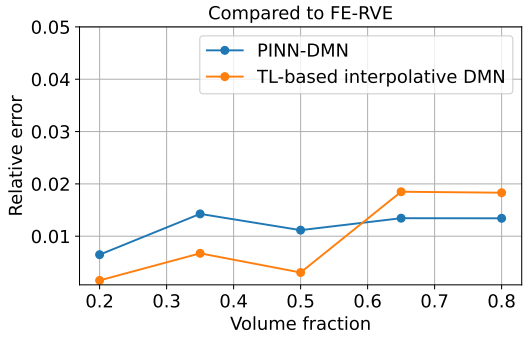


Figure 28: Relative error using real properties between the DMN predictions and the FE-RVE results obtained by PINN-DMN with the physical constraints and by the transfer-learning approach.

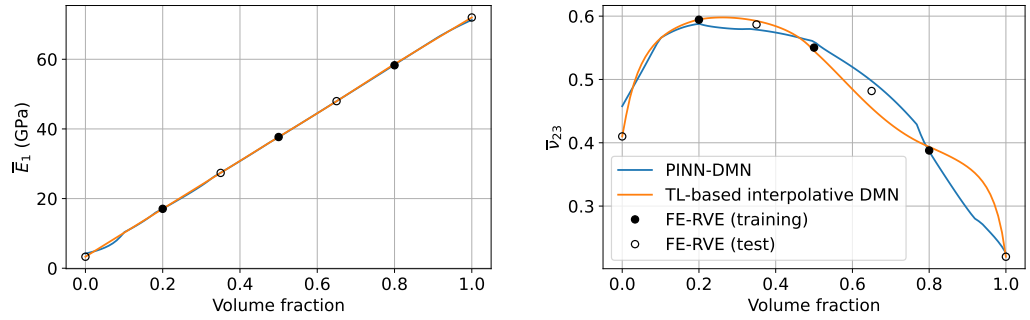
The elastic moduli prediction is also compared in Fig. 29 between these two approaches. Both models are able to capture the nonlinear influence of the volume fraction, especially on the in-plane Poisson ratio ν_{23}. While a perfect agreement is found for $\overline{E}_1$, the transfer-learning approach seems to slightly overfit the ν_{23} prediction between 0.5 and 1.0.

Similar to Fig. 24, the DMN weights evolution in the parametric space is also shown in Fig. 30 for the transfer-learning approach. Globally, 12 of the 32 material nodes are active. Due to transfer-learning and the ReLU activation function, the number of active nodes can only decrease [26]. While 12 material nodes are active for vf = 0.2 and vf = 0.5, only 9 active nodes are present for vf = 0.8. Compared to PINN-DMN, this gradual decrease of the active DMN material nodes may lead to a reduced expressive power. The functional dependence on vf is mainly realized by adapting weights values at each transfer-learning interpolation point. There is no gradual change in the ratios of active nodes for intermediate volume fractions.

Multiphysics property prediction

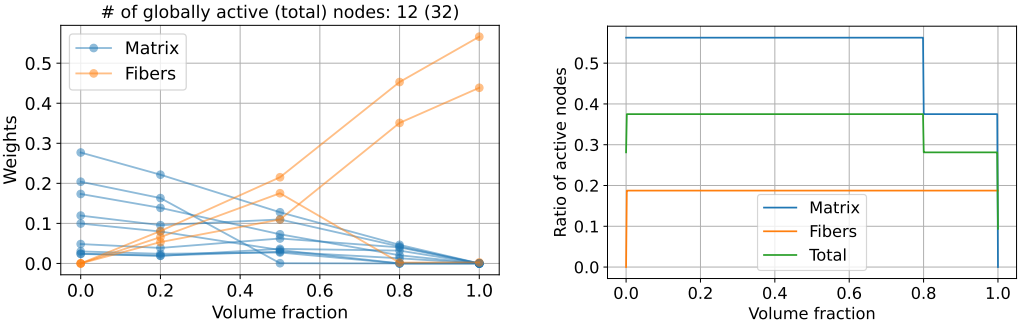
Using the proposed PINN-DMN architecture and the fitted parameters trained on isothermal linear elastic data, we will now predict the effective thermal conductivity $\overline{\boldsymbol{k}}$ and the effective CTE $\overline{\boldsymbol{\alpha}}$ with the redefinitions of the laminate homogenization function Lam in Sect. 2.3. The constituent properties used in this online prediction test are provided in Tab. 7. The effective properties predicted by PINN-DMN will be compared to FE-RVE simulation results, carried out using the Abaqus FE-RVE plugin [48].

In Fig. 31, the effective thermal conductivity $\overline{\boldsymbol{k}}$ predicted by PINN-DMN is compared with the FE-RVE results (or theoretical results for vf = 0 and vf = 1) as a function of the volume fraction. For both longitudinal and



(a) (b)

Figure 29: Homogenized elastic moduli prediction as a function of the fiber volume fraction obtained by PINN-DMN and the transfer-learning approach: (a) Young's modulus $\overline{E}_1$ and (b) Poisson ratio $\overline{\nu}_{23}$.



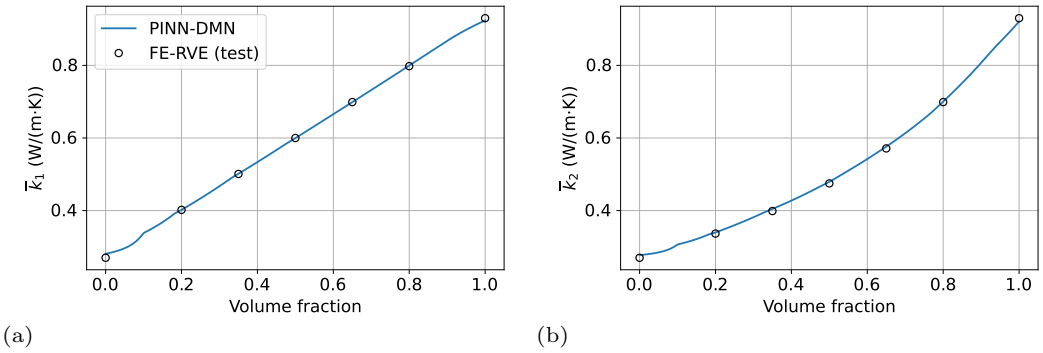
(a) (b)

Figure 30: Transfer-learning based interpolative DMN: (a) DMN weights evolution with the volume fraction; (b) variation of the ratio of active DMN nodes

	k (W/(m·K))	α (K^{-1})
Matrix	0.27	7×10^{-5}
Fiber	0.93	5×10^{-6}

Table 7: Real thermal conductivity and CTE properties of the two phases for the unidirectional fiber composite.

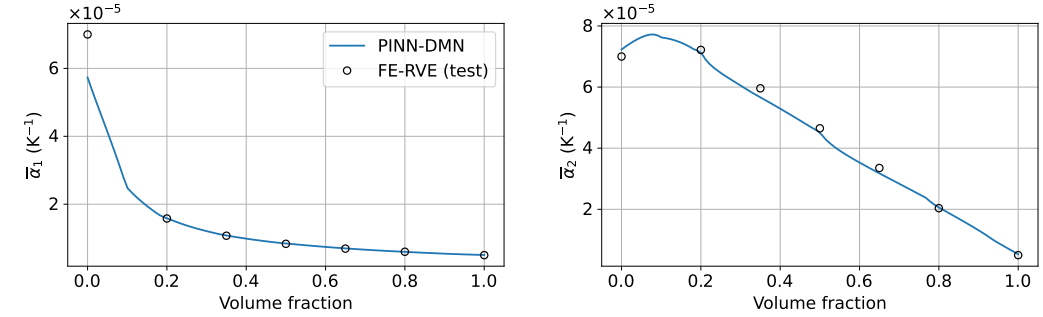
transverse components, an excellent agreement is obtained in the whole interval. This is a remarkable result knowing that our PINN-DMN is only trained using isothermal linear elastic data. This demonstrates that DMN learns the microstructure *per se*, and not a particular physics property in particular.



(a) (b)

Figure 31: Effective anisotropic thermal conductivity prediction (a) longitudinal $\overline{k}_1$ and (b) transverse $\overline{k}_2$.

The effective CTE is also computed in Fig. 32 using the trained PINN-DMN for an online prediction test. The nonlinear dependence of $\overline{\boldsymbol{\alpha}}$ on the volume fraction is well captured by PINN-DMN. A very good agreement with the FE-RVE data is also observed, even though the longitudinal $\overline{\alpha}_1$ is slightly underestimated near vf ≈ 0.



(a) (b)

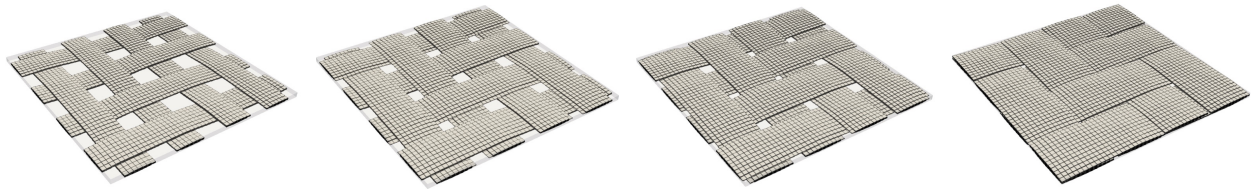
Figure 32: Effective anisotropic CTE prediction (a) longitudinal $\overline{\alpha}_1$ and (b) transverse $\overline{\alpha}_2$.

4.2. Woven composite

We now consider the 2×2 twill woven composites with varying tow volume fractions. Compared to the unidirectional fiber composite, now local material orientation is present for the yarns, illustrated in Fig. 10. The objective is to evaluate PINN-DMN for such microstructures in terms of elastic moduli prediction as a function of the volume fraction. Inverse identification of the material and microstructural parameters is also considered using PINN-DMN as an accurate and efficient surrogate of the parametric microstructure.

The 3-d finite element model is constructed using TexGen [50] with 4 volume fraction values for the tows, see Fig. 33. Three of them (vf $= 0.459$, vf $= 0.608$ and vf $= 0.729$) constitute the parametric sample set $\mathbb{P}$ and are used to generate training dataset, while vf $= 0.537$ is used to test interpolation accuracy. The FE-RVE analysis is conducted similarly as before under Abaqus to obtain the 6×6 linear elastic stiffness tensor. Each FE-RVE model contains 75000 voxel elements and 241899 degrees of freedom. The simulation is run with 24 threads using Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz and requires approximately 11 seconds.

The real linear elastic properties of the two phases are adapted from [18] and can be found in Tab. 8. The matrix is isotropic while the carbon fiber tows are assumed to be transversely isotropic in the local material frames. Compared to [18], the transverse Young's modulus E_2 of the tows is modified to satisfy transverse isotropy. A similar material sampling compared to Fig. 12 is performed to generate 500 input orthotropic material properties. Given the high moduli contrasts in the real properties, the synthetic yarn stiffnesses $\mathbb{C}_2$ are also generated to be statistically higher than the matrix ones $\mathbb{C}_1$. For training microstructures vf $= 0.459$, vf $= 0.608$ and vf $= 0.729$, 400 of the 500 samples are used as training dataset, while the other 100 are reserved for validation. For vf $= 0.537$, all the 500 samples are used for testing the interpolation accuracy.



(a) vf = 0.459 (b) vf = 0.537 (test) (c) vf = 0.608 (d) vf = 0.729

Figure 33: FE-RVE models for the woven composite with different fiber volume fractions. The models with vf = 0.459, vf = 0.608 and vf = 0.729 are used to generate training dataset, while vf = 0.537 is used to test interpolation accuracy.

	E (MPa)	ν
Matrix	3800	0.387

	E_1 (GPa)	E_2 (GPa)	ν_{12}	ν_{23}	μ_{12} (GPa)
Tow	78.8	6.24	0.35	0.6	2.39

Table 8: Real linear elastic properties of the two phases for the woven composite.

PINN-DMN is trained following Sect. 3.5 with the two physical constraints (28) and (35) and the ReLU activation function. In (35), the unidirectional material frame orientation tensor (32) is defined as the target for the matrix, since its material frame coincides with the global one. For the tows, we use the "planar isotropic" tensor (31) which characterizes the statistical spatial orientation of the yarns in the FE-RVE model. As before, the training process is realized 20 times with random initialization and the one with the least final loss value is chosen for further investigations. The median training time is summarized in Tab. 9.

	7 layers	9 layers
Median training time	800 s	1176 s

Table 9: Median training time with 7 or 9 DMN layers for the woven composite.

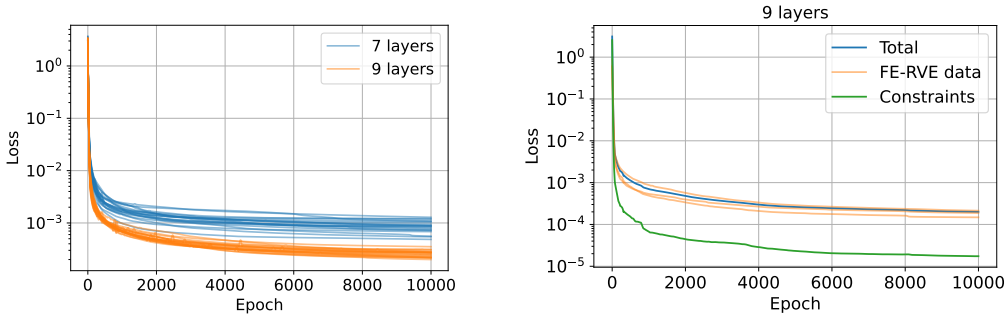
In Fig. 34(a), the loss histories are compared for 7 and 9 DMN layers. As for the unidirectional fiber composite, an increased expressive power is observed with more layers, leading to lower (final) loss values. The total loss can also be partitioned to a FE-RVE data part and a physical constraints part. These two parts are monotonically decreasing. The physical constraints (28) and (35) are well satisfied since the corresponding loss value is approaching 10^{-5} with 9 layers. In the sequel, we will report the results using 7 layers which provides satisfactory accuracy.

The volume fraction and the material frame orientation tensor for the tows are compared with their prescribed values in Fig. 35. The DMN volume fraction matches the FE-RVE data points and agrees well with the theoretical straight line even when evaluated outside the training region. In Fig. 35(b), the components $\mathsf{a}_{ii}^{(i)}$ of the DMN orientation tensor are shown. An excellent agreement is also obtained. Even without the additional rotations on the input layer, our DMN architecture is capable of learning local material orientation present in the microstructure.

In total, 81 DMN material nodes are globally active in the parametric space $0 \leq \text{vf} \leq 1$, as shown in Fig. 36. As in Fig. 24, the number of active nodes gradually decreases from 55% to 35% for the matrix while that of the tows increases from 50% to 65%. The overall ratio of the active nodes remains approximately 55% within the training region $[0.459, 0.729]$. This is believed to ensure sufficient expressive power for PINN-DMN.

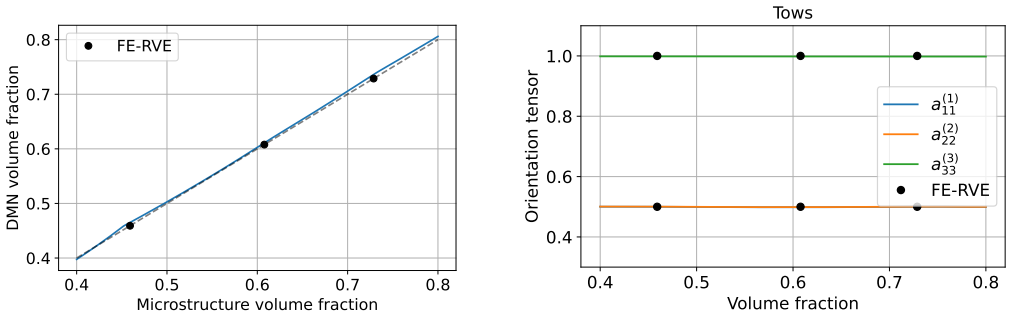
In Fig. 37, the training, validation and test errors are computed at different volume fractions. In Fig. 37(b), additional $2^7 = 128$ material rotations on the input layer are included. Even in the presence of local material orientation, such input material rotations do not significantly improve training and validation accuracy. Furthermore, the inclusion of their input rotations seems to overfit at the data points with much higher test interpolative errors. For PINN-DMN without these material rotations, the median error is approximately 2% for all the four microstructures. As in Fig. 18(a), the test error is only slightly larger compared to the training ones.

These training, validation and test errors are also presented for different anisotropies of the homogenized elasticity tensor, using (37) computed with the FE-RVE data. These errors are uniformly distributed and do not increase with anisotropy.



(a) (b)

Figure 34: PINN-DMN training for the woven composite: (a) loss histories with 7 and 9 DMN layers; (b) partition of the total loss into the FE-RVE data part and the physical constraints part.



(a) (b)

Figure 35: Verification of the physical constraints for the woven composite (a) volume fraction; (b) material frame orientation tensor for the tows.

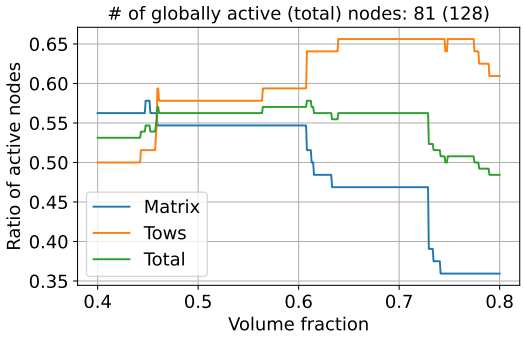
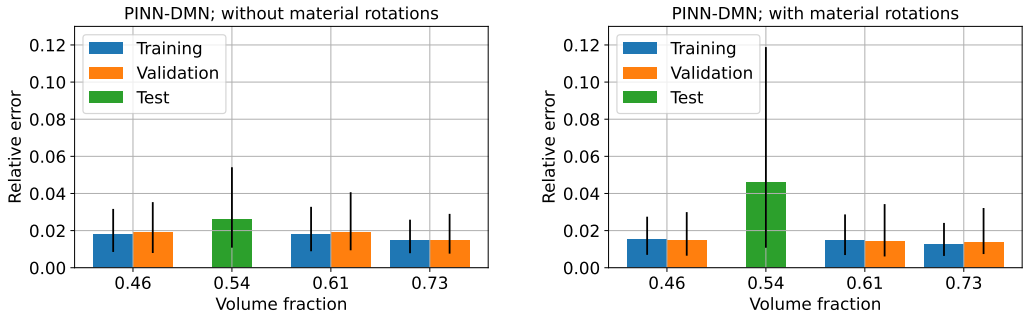


Figure 36: Variation of the ratio of active DMN nodes with varying volume fraction.



(a) (b)

Figure 37: Training, validation and test errors of PINN-DMN at different volume fractions represented by their respective 0.1, 0.5 (median) and 0.9-quantiles: (a) without material rotations; (b) with additional $2^7 = 128$ material rotations.

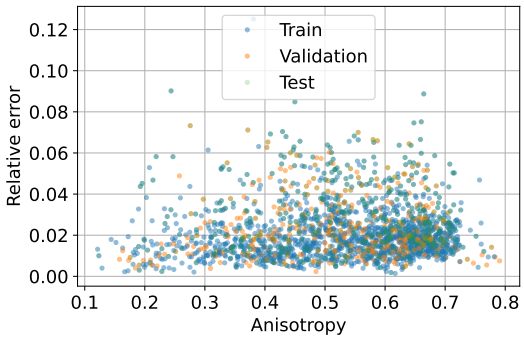
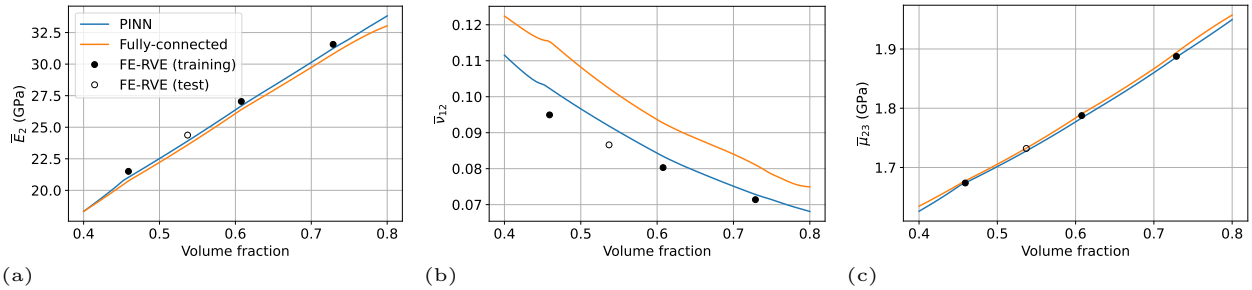


Figure 38: Training, validation and test errors for different output anisotropies.

With the real linear elastic properties provided in Tab. 8, the homogenized elastic moduli are computed in Fig. 39 with varying volume fraction. For comparison, the fully-connected architecture (23) is also tested with the physical constraints. Recall that (23) implies that the DMN rotations also become a function of the volume fraction. Both models capture well the nonlinear vf-dependence of these elastic moduli. The fully-connected architecture not only increases the number of fitting parameters ($\boldsymbol{\Theta}_1$ is now added), it does not improve prediction accuracy compared to PINN (25).



(a) (b) (c)

Figure 39: Homogenized elastic moduli prediction as a function of the tow volume fraction obtained by parametric DMN models: (a) in-plane Young's modulus $\overline{E}_2$, (b) in-plane Poisson ratio $\overline{\nu}_{12}$ and (c) transverse shear modulus $\overline{\mu}_{23}$. The FE-RVE training and test data are indicated.

The computational efficiency of DMN in terms of computing the homogenized elasticity tensor given $\mathbb{C}_1$ and $\mathbb{C}_2$ is compared to FE-RVE simulations in Tab. 10. Thanks to this significant speed-up, DMN can be further used for parametric analysis, uncertainty quantification [27] and material property calibration [17].

	FE-RVE	7-layer DMN
Wall time (speed-up)	11 s	6.62 ms (1662)

Table 10: Computational speed-up of DMN compared to FE-RVE in terms of homogenized elasticity tensor prediction. 24 cores of Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz are used in both cases.

Multiphysics property prediction

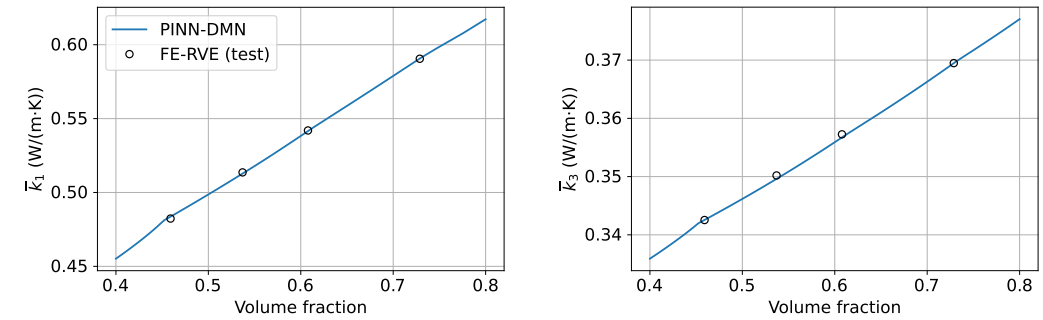
Using the previously trained 7-layer PINN-DMN, we will now predict the effective thermal conductivity and the effective CTE using the real properties of the constituents provided in Tab. 11. The effective properties predicted by PINN-DMN will be compared to FE-RVE simulation results, carried out using the Abaqus FE-RVE plugin [48].

The effective thermal conductivities predicted by PINN-DMN without additional training is given in Fig. 40. For both the in-plane and out-of-plane components, a very good agreement is found with the Abaqus FE-RVE results. Even though PINN-DMN is only trained using isothermal linear elastic data, it is capable of predicting other physical properties when the microstructure morphology varies.

In Fig. 41, the in-plane and out-of-plane effective CTE components predicted by PINN-DMN are compared with the Abaqus FE-RVE results. An excellent agreement is obtained in the given volume fraction range. Due to the

	k (W/(m·K))	α (K^{-1})
Matrix	0.3	7×10^{-5}
Tow (longitudinal)	1	4×10^{-6}
Tow (transverse)	0.4	4×10^{-5}

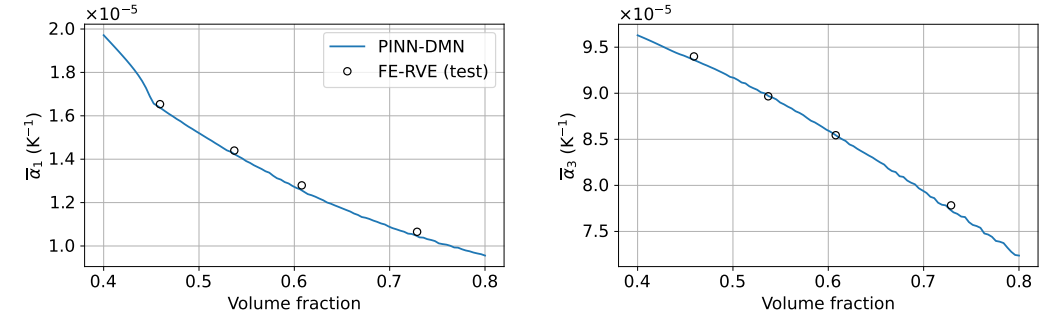
Table 11: Real thermal conductivity and CTE properties of the two phases for the woven composite.



(a) (b)

Figure 40: Effective anisotropic thermal conductivity prediction (a) in-plane $\overline{k}_1$ and (b) out-of-plane $\overline{k}_3$.

low CTE values of the tows (fibers) in the woven plane, the effective in-plane $\overline{\alpha}_1$ is well lower than the out-of-plane one.



(a) (b)

Figure 41: Effective anisotropic CTE prediction (a) in-plane $\overline{\alpha}_1$ and (b) out-of-plane $\overline{\alpha}_3$.

In this case, the second-order CTE tensor of the tows is transversely isotropic in its material frame, while the matrix remains isotropic. This implies that (22) must be applied as a neuron operation from the input layer to the output layer along with the effective stiffness tensor computation. If the laminate homogenization function of (22) for the effective CTE computation is naively applied directly to the obtained effective stiffness tensor, as shown in Fig. 42, the effective CTE would be incorrect.

Inverse identification of input material and microstructural parameters

Trained PINN-DMN can serve as an accurate and computationally efficient surrogate of the parametric microstructure. Not only it can be used in forward prediction of effective properties at different microstructural parameters, it can also be employed to identify both the *material* and *microstructural* properties in an inverse identification problem.

Now, the homogenized elasticity tensor $\overline{\mathbb{C}}$ is provided, and the objective is to identify the elastic properties of the matrix $\mathbb{C}_1$, those of the tows $\mathbb{C}_2$ as well as the volume fraction of the tows vf. In practice $\overline{\mathbb{C}}$ can be measured experimentally. Here we use the FE-RVE simulation result $\overline{\mathbb{C}}$ on the test microstructure vf $= 0.537$, obtained with the real properties in Tab. 8 which are now sought for.

Motivated by the fact that the gradients with respect to $(\mathbb{C}_1, \mathbb{C}_2, \text{vf})$ can be easily computed using automatic differentiation thanks to the PINN-DMN architecture, in this work we adopt a gradient-based optimization approach based on a loss function (40) which compares DMN prediction and the input homogenized data $\overline{\mathbb{C}}_{\text{Data}}$ with

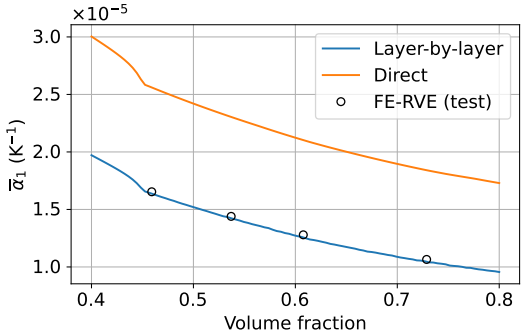


Figure 42: Comparison between a direct application and a DMN layer-by-layer application of the laminate homogenization function of (22) for the effective CTE computation.

the standard Frobenius norm. Similar to the training strategy of PINN-DMN, the Rprop optimizer is used for minimization.

$$\mathcal{L}_{\text{cal}} = \frac{\|\overline{\mathbb{C}}_{\text{DMN}} - \overline{\mathbb{C}}_{\text{Data}}\|^2}{\|\overline{\mathbb{C}}_{\text{Data}}\|^2}. \tag{40}$$

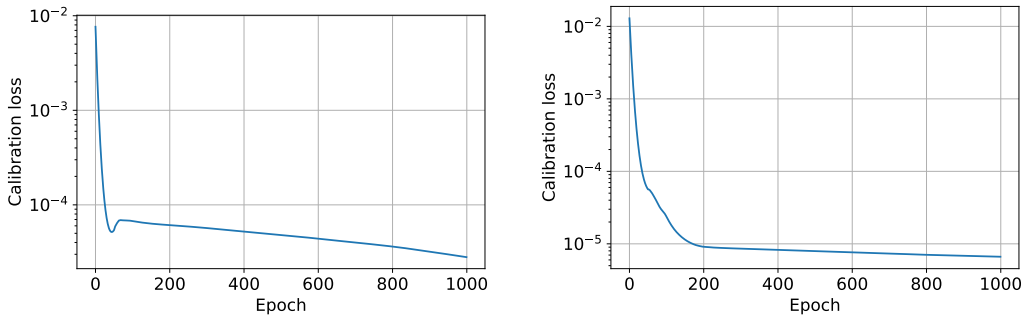
The optimization iterations require the initial guess values for the unknowns $(\mathbb{C}_1, \mathbb{C}_2, \text{vf})$. Hence, the real properties in Tab. 8 as well as the true volume fraction vf $= 0.537$ are randomly perturbed using a normal distribution with a coefficient of variation equal to 20%. The initial values generated by two random realizations are indicated in Tab. 12.

Matrix	E (MPa)	ν
Realization 1	4491	0.400
Realization 2	3908	0.437

Tow	vf	E_1 (GPa)	E_2 (GPa)	ν_{12}	ν_{23}	μ_{12} (GPa)
Realization 1	0.550	92.3	5.60	0.347	0.697	2.14
Realization 2	0.541	65.4	5.22	0.297	0.764	1.78

Table 12: Initial linear elastic properties of the two phases for the inverse identification problem.

The loss histories corresponding to these two initial values are given in Fig. 43. The loss function converges quickly and may reach 10^{-5} within 1000 optimization iterations. The input homogenized elasticity tensor $\overline{\mathbb{C}}$ is well recovered. According to Tab. 13, the relative error between the converged homogenized elasticity tensor and the input $\overline{\mathbb{C}}$ data is 0.42% and 0.26% for these two sets of initial values.



(a) (b)

Figure 43: Calibration loss histories using two initial values: (a) realization 1; (b) realization 2.

In Tab. 14, the inversely identified input parameters $(\mathbb{C}_1, \mathbb{C}_2, \text{vf})$ are reported. The material and microstructural parameters found with the realization 1 are similar to the actual properties in Tab. 8. However, with the realization

$\overline{\mathbb{C}}$	$E_1 = E_2$ (GPa)	E_3 (GPa)	ν_{12}	$\nu_{13} = \nu_{23}$	μ_{12} (GPa)	$\mu_{13} = \mu_{23}$ (GPa)	Error
Data	24.4	7.03	0.0866	0.558	1.87	1.73	
Realization 1	24.5, 24.3	7.05	0.0879	0.556, 0.555	1.87	1.73	0.42%
Realization 2	24.4	7.02	0.0863	0.560, 0.558	1.86	1.74	0.26%

Table 13: Converged homogenized elasticity tensor from two different realizations of initial values, compared with the true data.

2, the identified volume fraction 0.667 is higher than the data 0.537. Meanwhile, the longitudinal Young's modulus (64.6 GPa) of the tows is also smaller than the data (78.8 GPa). This illustrates the non-uniqueness of the inverse identification problem. Additional conditions may be provided by the user to further constrain the inverse problem. For instance, some input properties, like those of the matrix, can be assumed to be fixed. Nevertheless, such inverse identification problems can now be solved efficiently using PINN-DMN within seconds.

Matrix	E (MPa)	ν
Realization 1	4309	0.385
Realization 2	4624	0.375

Tow	vf	E_1 (GPa)	E_2 (GPa)	ν_{12}	ν_{23}	μ_{12} (GPa)
Realization 1	0.533	80.7	5.53	0.340	0.614	2.09
Realization 2	0.667	64.6	5.27	0.364	0.617	1.90

Table 14: Identified linear elastic properties of the two phases and the volume fraction for the inverse identification problem.

4.3. Ellipsoidal inclusion composite

Finally, we consider the ellipsoidal inclusion composite which contains two parameters: volume fraction vf of the fibers and the aspect ratio ar (ratio between the length and the diameter of the fiber). The aspect ratio parameter is purely geometrical and is *orthogonal* to vf. According to (24), we have thus $\boldsymbol{q} = (\text{ar})$ in this case. The objective is to evaluate PINN-DMN for such microstructures with multiple parameters with the linear elastic and nonlinear material behaviors.

The two-dimensional parametric space (vf, ar) is sampled using Sobol low-discrepancy sequence which ensures an optimal space-filling. Since it is deterministic, additional sampling points can thus be easily included. The bounds for these two parameters are

- Volume fraction: $[0.05, 0.065]$. Higher volume fractions would lead to mesh generation issues with the body-centered fiber packing.
- Aspect ratio: $[1, 100]$. Both spherical inclusions $\text{ar} = 1$ and slender fibers $\text{ar} \gg 1$ are covered with such large variation of this ar parameter.

In this work, 20 samples of $\boldsymbol{p}_i = (\text{vf}, \text{ar})_i$ are generated, see Fig. 44. The ar parameter is sampled in the log-scale. The first 10 points $\boldsymbol{p}_{0 \leq i < 10}$ constitute the parametric sample set $\mathbb{P}$ and are used to generate training dataset. The other 10 points $\boldsymbol{p}_{10 \leq i < 20}$ are reserved to evaluate the interpolation and extrapolation accuracy of PINN-DMN.

For each $\boldsymbol{p}_i = (\text{vf}, \text{ar})_i$, a 3-d FE-RVE model is built with the FE-RVE plugin [48] using the body-centered fiber packing. It is then solved under Abaqus to obtain the 6×6 linear elastic stiffness tensor. In Fig. 45, two examples are given corresponding to the first two sampling points. Each model contains approximately several hundreds of thousands of quadratic tetrahedral elements. Each run may take up to 60 seconds with 24 threads on Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz.

The real linear elastic properties of the two phases are the same as for the unidirectional fiber composite, given in Tab. 4. Given the high number of microstructures (10 for training, 10 for test), in total 150 input orthotropic material properties $(\mathbb{C}_1, \mathbb{C}_2)$ are generated. The synthetic fiber stiffnesses $\mathbb{C}_2$ are also generated to be statistically higher than the matrix ones $\mathbb{C}_1$. For the training microstructures $\boldsymbol{p}_{0 \leq i < 10}$, 100 of the 150 samples are used as training dataset, while the other 50 are reserved for validation. For the others $\boldsymbol{p}_{10 \leq i < 20}$, all the 150 material samples are used for testing the interpolation and extrapolation accuracy.

PINN-DMN is trained following Sect. 3.5 with the two physical constraints (28) and (35) and the ReLU activation function. Since local material orientation is absent, the unidirectional material frame orientation tensor

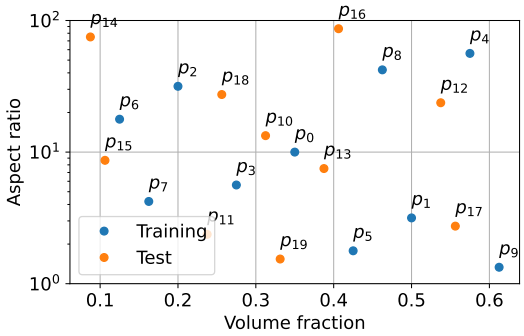
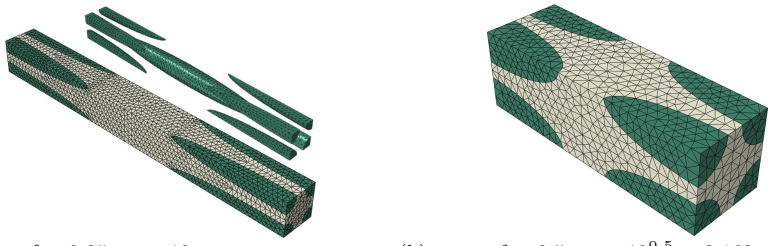


Figure 44: Sampling points in the parametric space for the ellipsoidal inclusion composite.



(a) $\boldsymbol{p}_0$: vf $= 0.35$, ar $= 10$ (b) $\boldsymbol{p}_1$: vf $= 0.5$, ar $= 10^{0.5} \approx 3.162$

Figure 45: Example FE-RVE models for the ellipsoidal composite with different fiber volume fraction and aspect ratio values: (a) $\boldsymbol{p}_0$; (b) $\boldsymbol{p}_1$.

(32) is defined as the target for both phases in (35). As before, the training process is realized 20 times with random initialization. The median training time is summarized in Tab. 15.

	7 layers	9 layers
Median training time	800 s	1198 s

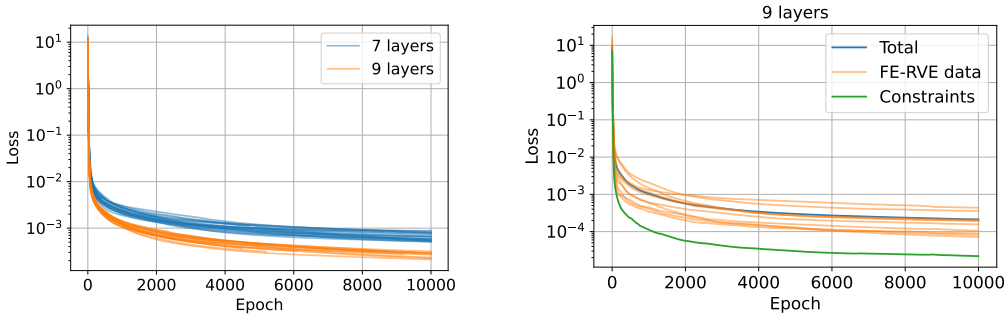
Table 15: Median training time with 7 or 9 DMN layers for the ellipsoidal inclusion composite.

The loss histories for 7 and 9 DMN layers are presented in Fig. 46(a). The final loss value is well decreasing with more layers. The total loss is then partitioned to a FE-RVE data part and a physical constraints part. These two parts are also monotonically decreasing. As before, the realization with the least final loss value is chosen for further investigations.

The variation of the number of active DMN nodes is presented separately for the matrix and for the fiber in Fig. 47. 7 DMN layers are used. In total, 73 out of the 128 DMN nodes are globally active in the parametric space $\boldsymbol{p} = (\text{vf}, \text{ar}) \in [0, 0.7] \times [1, 100]$. The volume fraction parameter determines the number of active matrix or fiber nodes while the aspect ratio parameter has practically no influence.

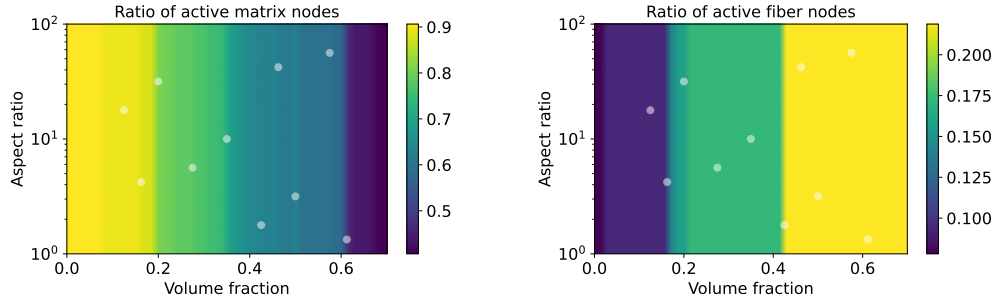
In Fig. 48(a), the training, validation and test errors are presented for PINN-DMN with 7 or 9 layers. These errors are first computed at each of the 20 sampling points, then aggregated using quantiles. All these errors become lower with more DMN layers. Overfitting is not observed since the validation errors are comparable with the training ones. For the training and validation errors, their statistical variations are limited. In the example of a 7-layer PINN-DMN, the 0.9-quantile error is less than 4% while the median error value is less than 2%. However, the test errors present larger statistical dispersion since their 0.9-quantiles reach approximately 10%. Due to a large amount of test points in the parametric space (see again Fig. 44), PINN-DMN is being evaluated both with interpolation and extrapolation. The 0.9-quantile of the test errors is mainly due to the presence of "outliers" at some particularly challenging test microstructures. The median test errors, nevertheless, are only slightly larger than the training and the validation errors, as before.

These errors are also computed for a fully-connected architecture (23) with 7 DMN layers. Compared to PINN (25), now the DMN fitting parameters become function of both microstructural parameters $\boldsymbol{w} = \boldsymbol{w}(\text{vf}, \text{ar})$ and $\boldsymbol{\theta} = \boldsymbol{\theta}(\text{vf}, \text{ar})$. The physical constraints (28) and (35) are included. According to Fig. 48(b), not only the fully-connected architecture does not significantly reduce training and validation errors, it also nearly doubles the



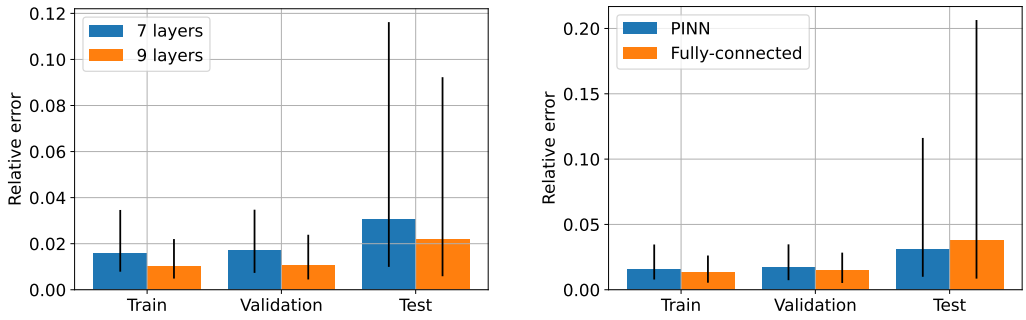
(a) (b)

Figure 46: PINN-DMN training for the ellipsoidal inclusion composite: (a) loss histories with 7 or 9 DMN layers; (b) partition of the total loss into the FE-RVE data part and the physical constraints part.



(a) (b)

Figure 47: Variation of the ratio of active DMN nodes in the parametric space: (a) matrix; (b) fiber



(a) (b)

Figure 48: Training, validation and test errors of parametric DMN models represented by their respective 0.1, 0.5 (median) and 0.9-quantiles: (a) comparison between 7 and 9 layers with the physics-informed architecture; (b) comparison with the fully-connected architecture using 7 layers.

0.9-quantile of the test errors. The PINN architecture (25) ensures hence a comparable expressive power in the parametric space with certain generalization capability.

In Fig. 49, the median errors are first computed at each of the 20 training and test sampling points, then interpolated and extrapolated to the whole parametric space using Kriging implemented in the SMT library [51]. In Fig. 49(a), the maximum error (10%) is localized at the "outlier" microstructure $\boldsymbol{p}_{16}$ presents a very high aspect ratio (87) compared to the training domain. Except this point, the median errors are between 1% and 4% and are visually uniform in the parametric space. Satisfying interpolation and extrapolation capabilities are also demonstrated. The fully-connected architecture, however, produces more error variations in the parametric space. This illustrates again the generalization capability of the PINN architecture.

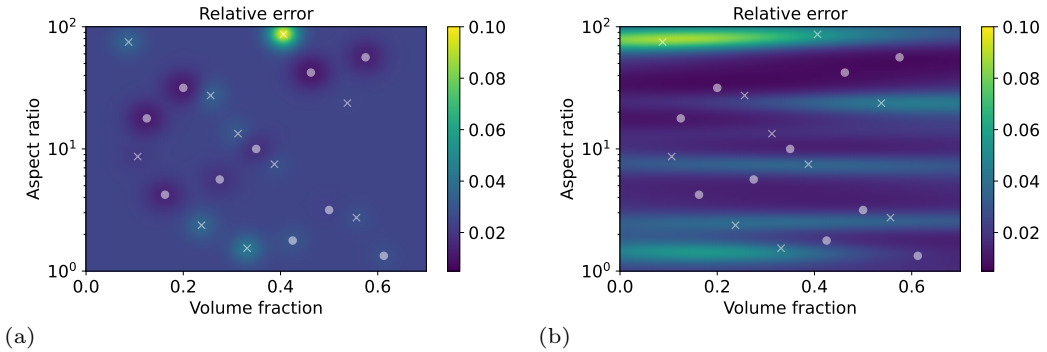


Figure 49: Median errors of parametric DMN models in the parametric space (a) PINN (b) fully-connected architecture.

With the real linear elastic properties in Tab. 4, the vf-dependence of the homogenized elastic moduli at fixed aspect ratio ar = 20 are illustrated in Fig. 50 using a 9-layer PINN-DMN. A perfect agreement is found with the additional test FE-RVE simulation data. In Fig. 51, the influence of the aspect ratio parameter is analyzed with a fixed vf = 0.4. Not only PINN-DMN is capable of predicting the increase of $\overline{E}_1$ with ar (at least in the training domain), it can also account for the more subtle variations of $\overline{\nu}_{12}$ and $\overline{\mu}_{23}$, which are all in good agreement with the FE-RVE data. Some deviations are found near the extrapolative test point ar = 100. While the physical constraints can indeed improve interpolative and extrapolative generalization capability of PINN-DMN, they cannot replace the input training data. If the prediction at higher aspect ratios is important, more corresponding FE-RVE data could be included during training.

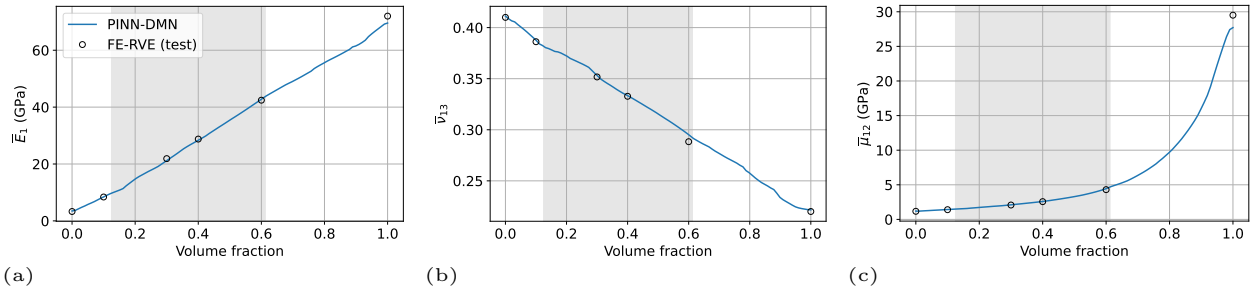
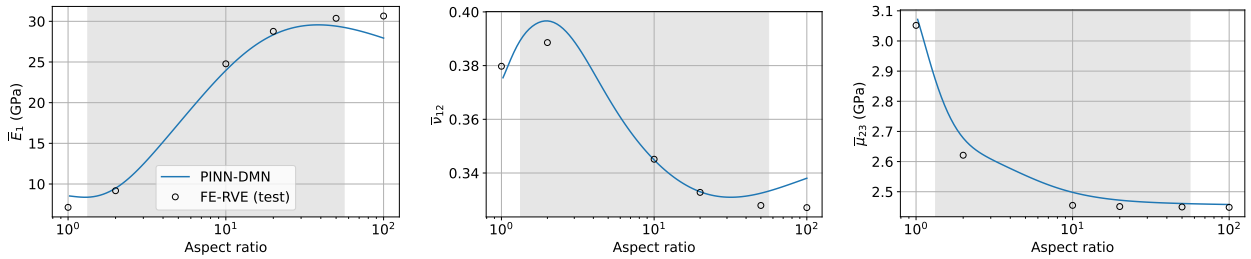


Figure 50: Homogenized elastic moduli prediction as a function of vf at fixed ar = 20 using a 9-layer PINN-DMN: (a) longitudinal Young's modulus $\overline{E}_1$, (b) transverse Poisson ratio $\overline{\nu}_{13}$ and (c) transverse shear modulus $\overline{\mu}_{12}$. The training domain is represented by a gray shadowed region. The FE-RVE test data and the theoretical values at vf = 0 and vf = 1 are also indicated.

Multiphysics property prediction

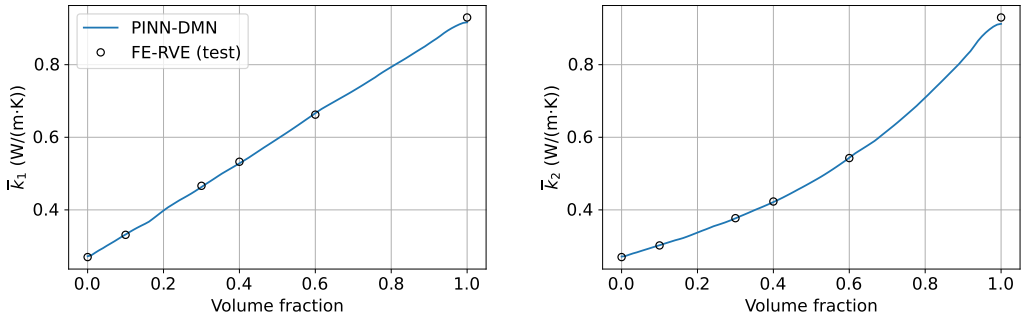
Using the real thermal conductivity and CTE properties of the constituents provided previously in Tab. 7, we will now use PINN-DMN trained with isothermal linear elastic data to predict the effective physical properties other than the mechanical ones. In Fig. 52 and Fig. 53, the vf- and ar-dependence of the effective thermal conductivity is respectively predicted and compared with the Abaqus FE-RVE results. A satisfying agreement is obtained for both the longitudinal and transverse components. This remarkable result illustrates again the multiphysics property prediction capability of DMN, knowing that the thermal conductivity data are not used as training data at all.

In Fig. 54 and Fig. 55, the effective anisotropic CTE is predicted while varying vf and ar. As before, a good agreement is found between the PINN-DMN prediction and the FE-RVE results.



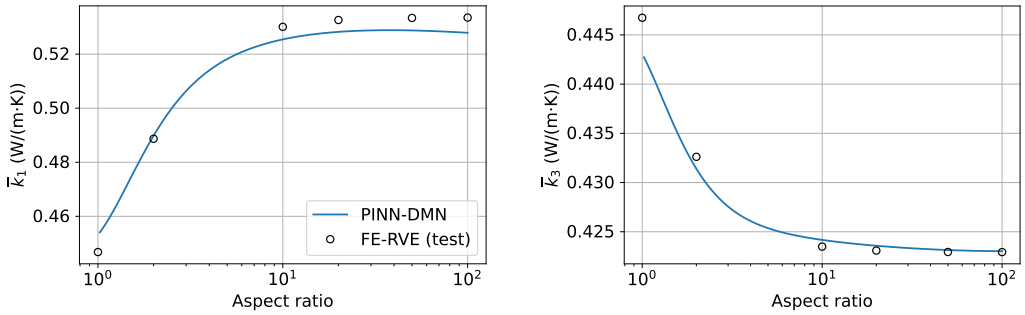
(a) (b) (c)

Figure 51: Homogenized elastic moduli prediction as a function of ar at fixed vf = 0.4 using a 9-layer PINN-DMN: (a) longitudinal Young's modulus $\overline{E}_1$, (b) transverse Poisson ratio $\overline{\nu}_{12}$ and (c) in-plane shear modulus $\overline{\mu}_{23}$. The training domain is represented by a gray shadowed region. The FE-RVE test data are also indicated.



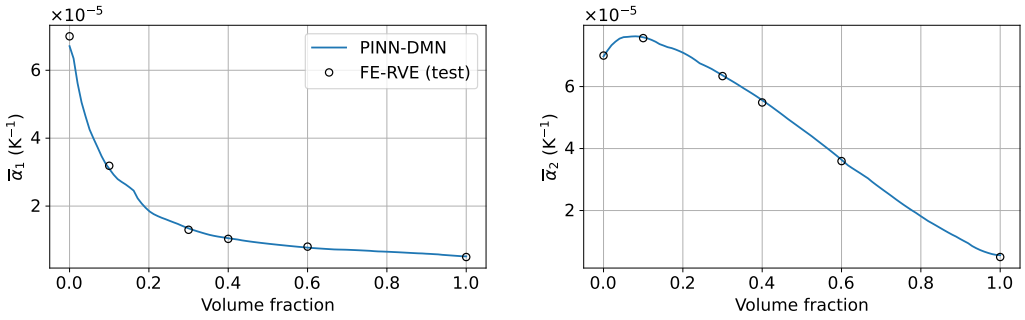
(a) (b)

Figure 52: Effective thermal conductivity prediction as a function of vf at fixed ar = 20 using a 9-layer PINN-DMN: (a) longitudinal $\overline{k}_1$ and (b) transverse $\overline{k}_3$.



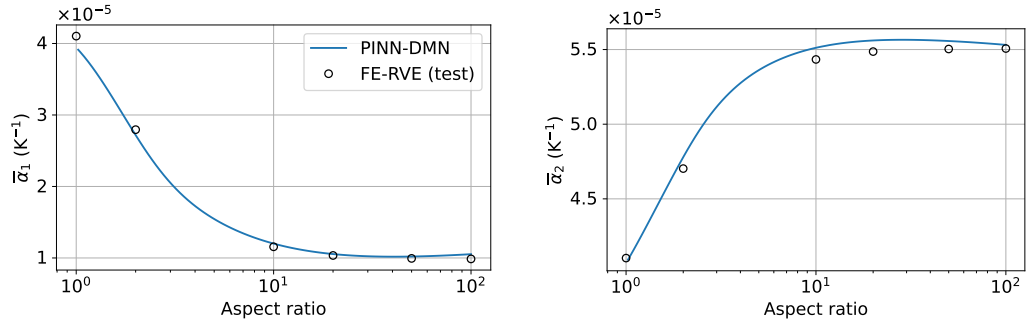
(a) (b)

Figure 53: Effective thermal conductivity prediction as a function of ar at fixed vf = 0.4 using a 9-layer PINN-DMN: (a) longitudinal $\overline{k}_1$ and (b) transverse $\overline{k}_3$.



(a) (b)

Figure 54: Effective anisotropic CTE prediction as a function of vf at fixed ar = 20 using a 9-layer PINN-DMN: (a) longitudinal $\overline{\alpha}_1$ and (b) transverse $\overline{\alpha}_2$.



(a) (b)

Figure 55: Effective anisotropic CTE prediction as a function of ar at fixed vf = 0.4 using a 9-layer PINN-DMN: (a) longitudinal $\overline{\alpha}_1$ and (b) transverse $\overline{\alpha}_2$.

Nonlinear simulations with an elasto-plastic matrix

The objective is now to evaluate PINN-DMNN with nonlinear material behaviors on different microstructural parameters. We assume that the polypropylene matrix follows an elasto-plastic law with an isotropic power-law hardening

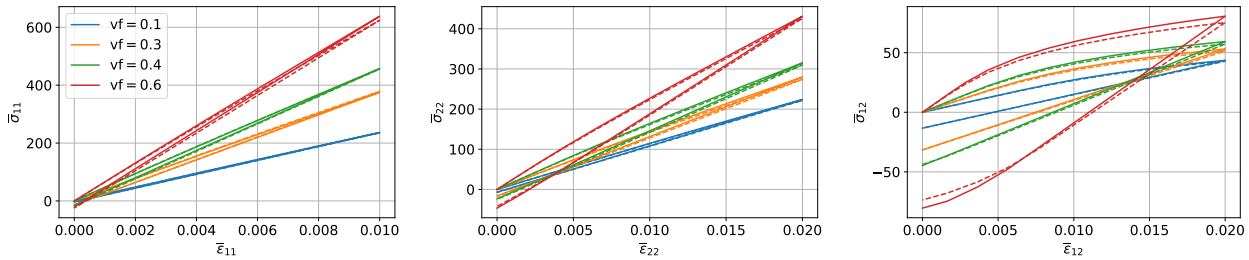
$$\sigma_{\mathrm{Y}} = \sigma_0 + k(p+\epsilon)^n,$$

where σ_{Y} defines the yield surface, σ_0 is the yield stress, and (k, n) describes strain hardening with the equivalent plastic strain p. The following numerical values are used: $\sigma_0 = 30$ MPa, $k = 293$ MPa and $n = 0.34$. The small value $\epsilon = 10^{-6}$ is used to avoid infinite derivative $\mathrm{d}\sigma_{\mathrm{Y}}/\mathrm{d}p$ at $p = 0$. The following triaxial strain loading–unloading path is considered

$$(\overline{\varepsilon}_{11}, \overline{\varepsilon}_{22}, \overline{\varepsilon}_{12}) = (0,0,0) \rightarrow (10^{-2}, 2\times10^{-2}, 2\times10^{-2}) \rightarrow (0,0,0).$$

The other strain components are assumed to be zero.

In Fig. 56, the corresponding stress–strain responses are presented (solid lines) in comparison with the FE-RVE simulation results (dashed lines), for different volume fraction values. The influence of the vf-parameter on the nonlinear effective responses is well captured by PINN-DMN. With increasing volume fraction, the maximum longitudinal effective stress $\overline{\sigma}_{11}$ increases from approximately 200 MPa to over 600 MPa. Furthermore, more plasticity effects are present on the $\overline{\sigma}_{22}$ and $\overline{\sigma}_{12}$ components. Good agreement between PINN-DMN and the FE-RVE results is observed.

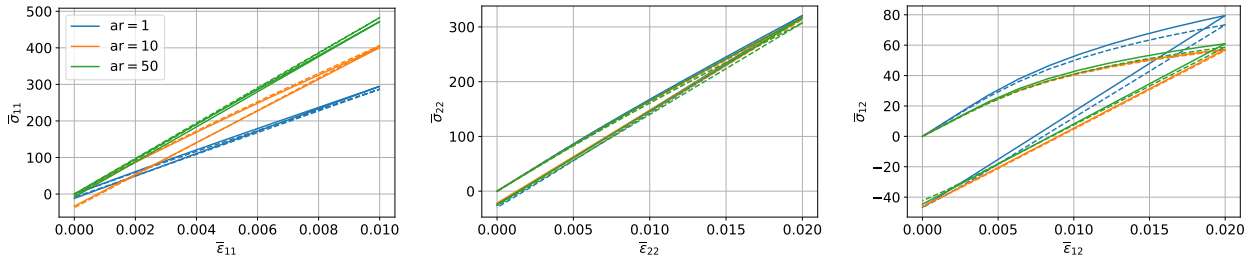


(a) (b) (c)

Figure 56: Effective nonlinear behavior of the ellipsoidal inclusion composite for different fiber volume fractions at fixed ar = 20 using a 9-layer PINN-DMN: (a) $\overline{\varepsilon}_{11} - \overline{\sigma}_{11}$; (b) $\overline{\varepsilon}_{22} - \overline{\sigma}_{22}$ and (c) $\overline{\varepsilon}_{12} - \overline{\sigma}_{12}$. The DMN predictions are represented by solid lines, while the FE-RVE data are dashed lines.

The ar-dependence of the nonlinear effective behaviors is then analyzed in Fig. 57. Similarly, the aspect ratio parameter has a bigger influence on the longitudinal effective stress, which increases from 300 MPa to nearly 500 MPa when ar varies from 1 (spherical inclusions) to 50 (slender fibers). The variation in the transverse tensile stress $\overline{\sigma}_{22}$ is negligible, while the transverse shear stress $\overline{\sigma}_{12}$ seems to converge within the given ar range. Again, PINN-DMN is in good agreement with FE-RVE responses.

Finally, we investigate the use of Aitken relaxation (15) on the convergence of the fixed-point problem (14). First, the tolerance value in (16) is analyzed in Fig. 58, with vf = 0.6 and ar = 20. For this problem, we observe that tol $= 10^{-1}$ is sufficient to obtain converged nonlinear responses.



(a) (b) (c)

Figure 57: Effective nonlinear behavior of the ellipsoidal inclusion composite for different fiber aspect ratios at fixed vf = 0.4 using a 9-layer PINN-DMN: (a) $\bar{\varepsilon}_{11} - \bar{\sigma}_{11}$; (b) $\bar{\varepsilon}_{22} - \bar{\sigma}_{22}$ and (c) $\bar{\varepsilon}_{12} - \bar{\sigma}_{12}$. The DMN predictions are represented by solid lines, while the FE-RVE data are dashed lines.

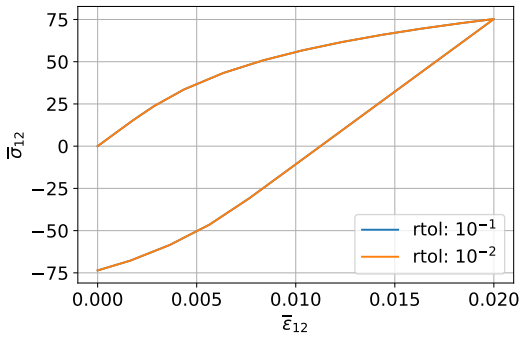
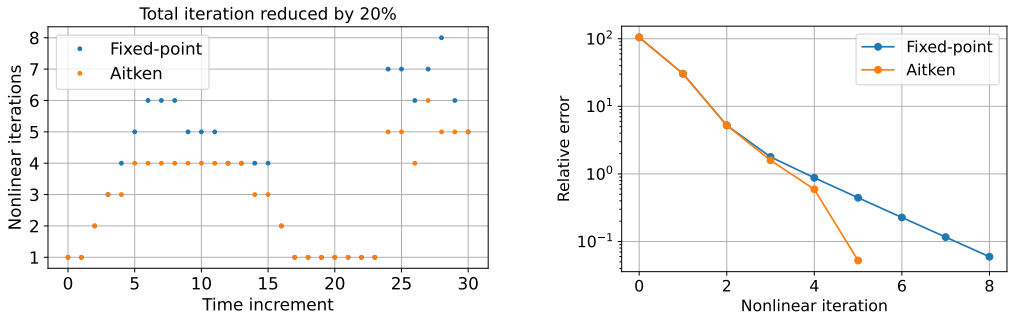


Figure 58: Effect of the tolerance value on the obtained nonlinear effective behavior for the ellipsoidal inclusion composite.

In Fig. 59(a), the number of fixed-point iterations are presented for each time increment. In (15), we use $\omega_0 = 1$, $\omega_{\min} = 1$ and $\omega_{\max} = 2$. This means we are over-relaxing the already converging fixed-point iterations. When the material behavior is linear, only one iteration is required. Otherwise, we observe an overall reduction in the number of iterations with Aitken relaxation. The total number of iterations is reduced by 20%, which leads to a reduction in the computational time by approximately 18% on our machine. This is due to the fact that Aitken relaxation essentially only involves a scalar product and a norm computation, and is hence much less costly than each fixed-point iteration.



(a) (b)

Figure 59: Influence of Aitken relaxation on the convergence of the nonlinear fixed-point problem: (a) number of fixed-point iterations for each time increment; (b) relative error history for the increment 28.

The history of the relative error (16) at time increment 28 is used to illustrate the effect of Aitken relaxation, in Fig. 59(b). The relaxation parameter begins to increase at iteration 3 and gets a value of 1.96 at iteration 4, which accelerates convergence of the fixed-point problem. Consequently, only 5 iterations are now required.

5. Conclusions

A novel physics-informed neural network-based parametric deep material network (PINN-DMN) architecture is proposed for heterogeneous materials with a varying microstructure described by multiple parameters. The dependence of the DMN fitting parameters on the microstructural ones is accounted for by a neural network informed by micromechanical properties (25). While the DMN weights only vary with the scalar volume fraction parameter (25a), the DMN rotations are influenced by other parameters that do not change the volume fraction (25b). Furthermore, two additional physical constraints (28) and (35) are prescribed so that PINN recovers the volume fraction and the material frame orientation tensor of the actual parametric microstructure. A unique offline training of PINN-DMN is required, based on a total loss function (38) that aggregates the FE-RVE data for different microstructural parameter values.

Based on the numerical results reported, the PINN architecture as well as the physical constraints reduce overfitting on the training data and improve interpolative and extrapolative generalization capabilities in the parametric space. The inclusion of initial material rotation matrices on the input layer does not significantly increase the expressive power even for microstructures with local material orientation. The linear and nonlinear effective behaviors of the parametric microstructure can be predicted by PINN-DMN. As an accurate and efficient surrogate, it can now be used in an inverse identification problem to calibrate input constituent properties and the microstructural parameters. We also show that the use of Aitken relaxation improves convergence of the fixed-point problem for online nonlinear simulations.

Even though hidden layers are ignored (25), our PINN-DMN also demonstrates satisfying expressive power for the parametric microstructures studied in this work. For microstructures with much more geometrical parameters, it can be expected that $\boldsymbol{q} \mapsto \boldsymbol{\theta}(\boldsymbol{q})$ in (25b) may become nonlinear and requires the composition of multiple affine and nonlinear activation layers [52]. It will be investigated in our future work.

DMN is also recast in a multiphysics setting. Through the redefinition of the laminate homogenization function, other physical properties such as the thermal conductivity and the coefficient of thermal expansion can be accurately predicted at the online prediction stage. This is a remarkable result since DMN is trained using the isothermal linear elastic data. The numerical results demonstrate that our PINN-DMN learns the *parametric* microstructure *per se*, and not a physics property in particular. Future work will be devoted to the offline training of PINN-DMN using multiphysics data.

Acknowledgments

The author thanks Rafael SALAZAR TIO for the fruitful discussions on using DMN in a multiphysics setting.

References

[1] P. Kanouté, D. P. Boso, J. L. Chaboche, B. A. Schrefler, Multiscale methods for composites: A review, Archives of Computational Methods in Engineering 16 (1) (2009) 31–75. doi:10.1007/s11831-008-9028-8.

[2] K. Matouš, M. G. Geers, V. G. Kouznetsova, A. Gillman, A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials, Journal of Computational Physics 330 (2017) 192–220. doi:10.1016/j.jcp.2016.10.070.

[3] J. Fish, G. J. Wagner, S. Keten, Mesoscopic and multiscale modelling in materials, Nature Materials 20 (6) (2021) 774–786. doi:10.1038/s41563-020-00913-0.

[4] C. L. Tucker III, E. Liang, Stiffness predictions for unidirectional short-fiber composites: Review and evaluation, Composites Science and Technology 59 (5) (1999) 655–671. doi:10.1016/s0266-3538(98)00120-1.

[5] O. Pierard, C. Friebel, I. Doghri, Mean-field homogenization of multi-phase thermo-elastic composites: a general framework and its validation, Composites Science and Technology 64 (10-11) (2004) 1587–1603. doi:10.1016/j.compscitech.2003.11.009.

[6] M. Geers, V. Kouznetsova, W. Brekelmans, Multi-scale computational homogenization: Trends and challenges, Journal of Computational and Applied Mathematics 234 (7) (2010) 2175–2182. doi:10.1016/j.cam.2009.08.077.

[7] Z. Liu, C. Wu, M. Koishi, A deep material network for multiscale topology learning and accelerated nonlinear modeling of heterogeneous materials, Computer Methods in Applied Mechanics and Engineering 345 (2019) 1138–1168. doi:10.1016/j.cma.2018.09.020.

[8] S. Kumar, D. M. Kochmann, What machine learning can do for computational solid mechanics, in: Current Trends and Open Problems in Computational Mechanics, Springer International Publishing, 2022, pp. 275–285. doi:10.1007/978-3-030-87312-7_27.

[9] Z. Liu, C. Wu, Exploring the 3d architectures of deep material network in data-driven multiscale mechanics, Journal of the Mechanics and Physics of Solids 127 (2019) 20–46. doi:10.1016/j.jmps.2019.03.004.

[10] S. Gajek, M. Schneider, T. Böhlke, On the micromechanics of deep material networks, Journal of the Mechanics and Physics of Solids 142 (2020) 103984. doi:10.1016/j.jmps.2020.103984.

[11] S. Gajek, M. Schneider, T. Böhlke, An FE-DMN method for the multiscale analysis of short fiber reinforced plastic components, Computer Methods in Applied Mechanics and Engineering 384 (2021) 113952. doi:10.1016/j.cma.2021.113952.

[12] S. Gajek, M. Schneider, T. Böhlke, An FE-DMN method for the multiscale analysis of thermomechanical composites, Computational Mechanics 69 (5) (2022) 1087–1113. doi:10.1007/s00466-021-02131-0.

[13] H. Wei, C. T. Wu, W. Hu, T.-H. Su, H. Oura, M. Nishi, T. Naito, S. Chung, L. Shen, LS-DYNA machine learning-based multiscale method for nonlinear modeling of short fiber-reinforced composites, Journal of Engineering Mechanics 149 (3) (mar 2023). doi:10.1061/jenmdt.emeng-6945.

[14] Z. Liu, Deep material network with cohesive layers: Multi-stage training and interfacial failure analysis, Computer Methods in Applied Mechanics and Engineering 363 (2020) 112913. doi:10.1016/j.cma.2020.112913.

[15] Z. Liu, Cell division in deep material networks applied to multiscale strain localization modeling, Computer Methods in Applied Mechanics and Engineering 384 (2021) 113914. doi:10.1016/j.cma.2021.113914.

[16] A. P. Dey, F. Welschinger, M. Schneider, S. Gajek, T. Böhlke, Training deep material networks to reproduce creep loading of short fiber-reinforced thermoplastics with an inelastically-informed strategy, Archive of Applied Mechanics 92 (9) (2022) 2733–2755. doi:10.1007/s00419-022-02213-2.

[17] A. P. Dey, F. Welschinger, M. Schneider, S. Gajek, T. Böhlke, Rapid inverse calibration of a multiscale model for the viscoplastic and creep behavior of short fiber-reinforced thermoplastics based on deep material networks, International Journal of Plasticity 160 (2023) 103484. doi:10.1016/j.ijplas.2022.103484.

[18] L. Wu, L. Adam, L. Noels, Micro-mechanics and data-driven based reduced order models for multi-scale analyses of woven composites, Composite Structures 270 (2021) 114058. doi:10.1016/j.compstruct.2021.114058.

[19] V. D. Nguyen, L. Noels, Micromechanics-based material networks revisited from the interaction viewpoint: Robust and efficient implementation for multi-phase composites, European Journal of Mechanics - A/Solids 91 (2022) 104384. doi:10.1016/j.euromechsol.2021.104384.

[20] V. D. Nguyen, L. Noels, Interaction-based material network: A general framework for (porous) microstructured materials, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114300. doi:10.1016/j.cma.2021.114300.

[21] B. Halphen, Q. S. Nguyen, Sur les matériaux standard généralisés, Journal de Mécanique 14 (1) (1975) 39–63.

[22] G. Allaire, Shape Optimization by the Homogenization Method, Springer New York, 2002. doi:10.1007/978-1-4684-9286-6.

[23] G. W. Milton, The Theory of Composites, Cambridge University Press, 2002. doi:10.1017/cbo9780511613357.

[24] Z. Hashin, S. Shtrikman, On some variational principles in anisotropic and nonhomogeneous elasticity, Journal of the Mechanics and Physics of Solids 10 (4) (1962) 335–342. doi:10.1016/0022-5096(62)90004-2.

[25] G. A. Francfort, F. Murat, Homogenization and optimal bounds in linear elasticity, Archive for Rational Mechanics and Analysis 94 (4) (1986) 307–334. doi:10.1007/bf00280908.

[26] Z. Liu, C. T. Wu, M. Koishi, Transfer learning of deep material network for seamless structure-property predictions, Computational Mechanics 64 (2) (2019) 451–465. doi:10.1007/s00466-019-01704-4.

[27] T. Huang, Z. Liu, C. Wu, W. Chen, Microstructure-guided deep material network for rapid nonlinear material modeling and uncertainty quantification, Computer Methods in Applied Mechanics and Engineering 398 (2022) 115197. doi:10.1016/j.cma.2022.115197.

[28] S. G. Advani, C. L. Tucker, The use of tensors to describe and predict fiber orientation in short fiber composites, Journal of Rheology 31 (8) (1987) 751–784. doi:10.1122/1.549945.

[29] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. doi:10.1016/j.jcp.2018.10.045.

[30] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Reviews Physics 3 (6) (2021) 422–440. doi:10.1038/s42254-021-00314-5.

[31] J. Diebel, Representing attitude: Euler angles, unit quaternions, and rotation vectors, Matrix 58 (15-16) (2006) 1–35.

[32] I. Doghri, L. Adam, N. Bilger, Mean-field homogenization of elasto-viscoplastic composites based on a general incrementally affine linearization method, International Journal of Plasticity 26 (2) (2010) 219–238. doi:10.1016/j.ijplas.2009.06.003.

[33] I. Ramière, T. Helfer, Iterative residual-based vector methods to accelerate fixed point iterations, Computers and Mathematics with Applications 70 (9) (2015) 2210–2226. doi:10.1016/j.camwa.2015.08.025.

[34] B. M. Irons, R. C. Tuck, A version of the aitken accelerator for computer iteration, International Journal for Numerical Methods in Engineering 1 (3) (1969) 275–277. doi:10.1002/nme.1620010306.

[35] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, B. Uekermann, preCICE – a fully parallel library for multi-physics surface coupling, Computers and Fluids 141 (2016) 250–258. doi:10.1016/j.compfluid.2016.04.003.

[36] Y. Benveniste, G. J. Dvorak, On a correspondence between mechanical and thermal effects in two-phase composites, in: Micromechanics and Inhomogeneity, Springer New York, 1990, pp. 65–81. doi:10.1007/978-1-4613-8919-4_4.

[37] J. K. Bauer, T. Böhlke, Variety of fiber orientation tensors, Mathematics and Mechanics of Solids 27 (7) (2021) 1185–1211. doi:10.1177/10812865211057602.

[38] V. Müller, T. Böhlke, Prediction of effective elastic properties of fiber reinforced composites using fiber orientation tensors, Composites Science and Technology 130 (2016) 36–45. doi:10.1016/j.compscitech.2016.04.009.

[39] J. K. Bauer, M. Schneider, T. Böhlke, On the phase space of fourth-order fiber-orientation tensors, Journal of Elasticity (jan 2023). doi:10.1007/s10659-022-09977-2.

[40] T. Böhlke, Application of the maximum entropy method in texture analysis, Computational Materials Science 32 (3-4) (2005) 276–283. doi:10.1016/j.commatsci.2004.09.041.

[41] T. Lee, M. Leok, N. H. McClamroch, Global symplectic uncertainty propagation on SO(3), in: 2008 47th IEEE Conference on Decision and Control, IEEE, 2008. doi:10.1109/cdc.2008.4739058.

[42] M. D. Mckay, R. J. Beckman, W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61. doi:10.1080/00401706.2000.10485979.

[43] A. Norris, The isotropic material closest to a given anisotropic material, Journal of Mechanics of Materials and Structures 1 (2) (2006) 223–238. doi:10.2140/jomms.2006.1.223.

[44] I. M. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, USSR Computational Mathematics and Mathematical Physics 7 (4) (1967) 86–112. doi:10.1016/0041-5553(67)90144-9.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, Curran Associates Inc., Red Hook, NY, USA, 2019.

[46] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T. P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima, in: International Conference on Learning Representations, 2017.

[47] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: IEEE International Conference on Neural Networks, IEEE, 1993. doi:10.1109/icnn.1993.298623.

[48] R. McLendon, Micromechanics Plugin for Abaqus, https://www.linkedin.com/pulse/micromechanics-plugin-abaqus-ross-mclendon/, accessed: 2023-02-13 (2017).

[49] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2014). doi:10.48550/ARXIV.1412.6980.

[50] L. Brown, A. Long, Modeling the geometry of textile reinforcements for composites: TexGen, in: Composite Reinforcements for Optimum Performance, Elsevier, 2021, pp. 237–265. doi:10.1016/b978-0-12-819005-0.00008-3.

[51] M. A. Bouhlel, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, J. R. Martins, A python surrogate modeling framework with derivatives, Advances in Engineering Software 135 (2019) 102662. doi:10.1016/j.advengsoft.2019.03.005.

[52] J. He, L. Li, J. Xu, C. Zheng, Relu deep neural networks and linear finite elements, Journal of Computational Mathematics 38 (3) (2020) 502–527. doi:https://doi.org/10.4208/jcm.1901-m2018-0160.
URL http://global-sci.org/intro/article_detail/jcm/15798.html